VIGNAN'S

Foundation for Science, Technology & Research

(Deemed to be University)

-Estd. u/s 3 of UGC Act 1956

R22

ACADEMIC REGULATIONS, CURRICULUM AND COURSE CONTENTS FOR

B. TECH.

FOUR YEAR DEGREE PROGRAMME IN THE SPECIALIZATION OF

INFORMATION TECHNOLOGY

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be UNIVERSITY)
-Estd. u/s 3 of UGC Act 1956

R22 Academic Regulations

In Compliance with NEP 2020

Prof. P. NAGABHUSHAN
B.E., M.Tech., Ph.D., FIE, FIETE, FIAPS.
Vice Chancellor, VFSTR
Life time Professor, IIIT – Allahabad, Prayagraj
(Formerly: Director, IIIT-Allahabad 2017-22)

PREFACE

'You are born to Blossom' – What an inspiring title the book authored by APJ Abdul Kalam and Arun K Tiwari carries. The journey to blossom has got to be heralded by education. The purpose of education is to ensure that the 'Life Blossoms'. Earning a degree and getting a placement should be the just happening things, and should not become the only celebrated goals for education. In the book cited above, Honourable Kalam, Former President of India, underscores that "The scheme of civil society depends on Educating young people to become enlightened citizens and adults who are responsible, thoughtful and enterprising"

VIGNAN aims to seed these concepts in every learner who transits through this temple of learning. The doctrine of VIGNAN entitled R-22 contains the principles of policies laid down by the University, to realize the spirit of "Blossoming the lives" providing a foundation-strong professional education on the ethos of 'Creative learning for Critical thinking and Critically analysing for Creative decision making'. Certainly, our University is one of the earliest Universities, in fact the University is a trend setting one in completely internalising the concepts of the policies brought out in National Education Policy (New Educational Policy) NEP-2020, and inculcating the spirit in R-22. The R-22 document articulates the Academic Regulations of the University, which is being presented now and shall be inforce with immediate effect from the academic year 2022-23, not only for those who have joined in 2022, also the aspirants of 2021-22 are enabled into the navigation.

R-22 presents a novel design for the academic pursuit, making an exploratory cross disciplinary traversal for a learner who should find learning both holistic and experiential. The learner is ensured to enjoy the continuity in learning and the learner is supported to align and realign, enroute utilising the benefits of constructive feedbacks that s/he receives because of continuous assessment. S/he will be empowered to enjoy the opportunities to explore, experiment and experience.

R-22 eliminates the melancholy of examinations. The expected severity of breakdown due to the anxiety of examination system is replaced by an affectionate assessment system, increasing the effectiveness in accomplishing the outcomes.

In brief, NEP-2020 compliant revised academic regulation of the University – the R-22, is VIGNAN's commitment to alleviate the acuteness in the present educational practices. It intends to provide a strategic solution to the critical observation made by Bharat ratna awardee, Professor. CNR Rao – "India has exam system, not education system. When will young people stop taking exams and do something worthwhile?" (Thought for the Day, Times of India 13.08.2022)

Here is R-22, which assures that the learners at VIGNAN are bound to do something worthwhile – very much worthwhile.

***

REGULATIONS - TABLE OF CONTENTS

* Programme will be offered based on satisfactory strength of students willing to register, after receiving the formal AICTE approval.

Academic Regulations, Curriculum and Course Contents

EXECUTIVE ABSTRACT

R22 - Academic regulations, Curriculum and course contents, is an articulation of the VFSTR deemed to be University's commitment towards NEP-2020, with a view that it enables student(s) to maintain the spirit of continuous learning and continuous assessment to replace the normal tendency of preparing just before a test or an examination. The proposed framework accomplishes multi-disciplinary holistic education, continuous assessment along with multiple honorable exit options if a student falls short to complete the requirements to earn the degree within the stipulated period including the permissible spill over period.

R22 is oriented towards multi-disciplinary holistic education at the undergraduate level that includes integrated and rigorous exposure to science, humanities, management, and professional domains, as well as sufficient flexibility in curricular structures that allow students to choose electives from the same and other disciplines. Such holistic and diverse education will assist the candidate in transforming into all-rounded persons. Similarly, in line with NEP-2020, more weight will be given to continuous/ formative assessment, which is an Integrated learning model comprising Learning – Thinking – Understanding – Skilling – Applying – Creating. Emphasis on continuous formative assessment with a creative summative assessment will facilitate the candidate to "Move away from high stake examinations – towards more continuous and comprehensive evaluation".

The B.Tech. degree offered will be for four years' **(8 semesters) duration** with lateral entry and exit options within this period, with suitable certifications that will enable the candidate to have a professional career and as well as serve as a reminder to return and update his / her qualification in the future. Following three years of study and the completion of the required credits, an Engineering diploma or a Bachelor's degree will be conferred. However, the intention of the learners is not to join for the award of the diploma / degree with lateral exit, but to acquire a B.Tech. degree which offers add-on certifications such as Honours/ Minor/ Add-on Diploma.

Salient features of the regulation

- Multidisciplinary
- Continuous learning
- Continuous assessment
- Honours/ Research Honours / Minor / Add-on Diploma / Add-on Certification
- Lateral entry and lateral exit options
- Sabbatical Semester Drop option to pursue innovation, incubation, entrepreneurial and advanced exploratory activities and subsequent re-entry
- Dual B.Tech. + M. Tech. / MBA degree of 5 years
- Credit earning by credit transfer

1 INTRODUCTION

This document contains the academic regulations, scheme of assessments, curriculum, detailed syllabi, course contents with text / reference books recommended, course outcomes, skills acquired and the projects / assignments that are to be performed for each course for the conduct of 4-year B.Tech. degree programmes. The various B.Tech. degree programmes under different schools in VFSTR are as listed below. The 2-character codes indicated in parentheses are their branch discipline codes.

I. **School of Agriculture and Food Technology**

- Agriculture Engineering (AG)
- Food Technology (FT)

II. **School of Biotechnology and Pharmaceutical Sciences**

- Bioinformatics (BI)
- Biotechnology (BT)

III. **School of Computing and Informatics**

- Computer Science and Engineering (CS)
- Computer Science and Engineering - Artificial Intelligence and Machine Learning (AI)
- Computer Science and Business Systems (CB)
- Computer Science and Engineering - Cyber Security (CY)
- Computer Science and Engineering – Data Science (DS)
- Information Technology (IT)

IV. **School of Core Engineering**

- Chemical Engineering (CH)
- Civil Engineering (CE)
- Mechanical Engineering (ME)
- Robotics and Automation (RA)
- Textile Technology (TT)

V. **School of Electrical, Electronics and Communication Engineering**

- Biomedical Engineering (BM)
- Electrical and Electronics Engineering (EE)
- Electronics and Communication Engineering (EC)

1.1 Definition

For the purpose of R22 regulation, definitions as follows shall apply:

- **“Degree”** shall refer to the B.Tech. Degree Program.
- **“Course”** shall refer to such Course(s) for which a student shall earn Credits after due assessment as per the laid provisions. Project is also treated as a Course.
- **“Academic activities”** shall refer to the activities like Lecture (Physical Lecture Session), Tutorial (Participatory discussion / Self-Study / Desk Work / Quiz / Seminar Presentation, *etc* activities that make the student absorb & assimilate, the delivered contents effectively) and Practical / Practice sessions (includes Hands on Experience / Lab experiments / Field Studies / Case Studies *etc* activities that enable the student to acquire the requisite skill).
- **“Continuous Assessment”** shall refer to the assessment of the student spread over the entire semester on the various constituent components of the prescribed course.
- **“Semester”** shall refer to a period covering the two assessment periods *viz* Formative and Summative Assessment period. A semester would generally be spread over twenty weeks.
- **“Course Drop”** shall refer to a student having to undertake a repeat of the Course(s) not being able to complete the Credit requirements of the Course(s), under the conditions stipulated in the regulation.
- **“Supplementary Examinations”** shall refer to the examination(s) conducted to allow the student to appear in the un-cleared Semester - End summative assessment component.
- **“Blank Semester”** shall refer to a Semester in which a student either does not register for any course at the beginning of the Semester OR chooses to DROP all courses OR is so compelled to DROP all the courses, as the case may be.
- **“Semester Drop”** shall refer to availing a blank semester. However, if drop is availed to pursue a creative extension activity, then it is defined as **semester sabbatical**.

- **“Spill Over Semester”** shall refer to the additional semester(s) beyond the completion of prescribed normal semesters.
- **“AAA Section”** shall refer to the Academics, Assessment and Award Section of the Institute.
- **“Attendance”** refers to the Physical personal presence in an academic activity session.
- **“Summer Semester”** refers to a Semester that is scheduled to be held during the intervening period of Even and Odd Semester (*i.e*. Summer Vacation period).
- **“Themes”** refer to the courses offered in a particular stream other than offered by the regular departments, for example NCC, Entrepreneurship, Fitness and Living, *etc*.
- **“School”** refers to a division of institute dealing with two or more specific areas of discipline / study comprising of the departments related with exclusive emphasis on trans-disciplinary research.
- **“Department”** refers to a division of institute dealing with a specific area of discipline / study.
- **“HoD”** refers to the Head of the respective Department, where the student is enrolled for his / her Branch of Study.
- **“Center”** refers to a structured unit within the school / department established with the purpose to carry out advanced research.
- **“Grade Point”** refers to the quantification of the performance of a candidate in a particular course as defined herein.
- **“SGPA”** refers to the Semester Grade Point Average and is calculated as detailed in the regulations subsequently.
- **“CGPA”** refers to the Cumulative Grade Point Average and is calculated as detailed in the regulations subsequently.
- **“Division”** refers to the Division awarded to the student as per the mechanism detailed in the regulations subsequently.
- **“Internship”** refers to onsite Practical Training offered by reputed companies / Institutions, in India or abroad. To be undertaken with (or seeking) prior approval of the respective HoD.
- **“Project”** refers to a course executed by a candidate on a specific research problem at VFSTR / any organization of repute. To be undertaken with (or seeking) prior approval of the respective HoD.
- **“Credit equivalence and credit transfer committee”** refers to the committee designated to look into for credit equivalence and credit transfer.
- **“Honorable Exit Option”** refers to the Exit Options available to students, when they are unable to complete the prescribed four-year B.Tech. Degree program in seven successive years.

1.2 Academic Administration

The academic programmes of VFSTR are governed by the rules and regulations approved by the Academic Council from time to time. The various academic activities are conducted following a fixed time schedule duly approved by the Academic Council in line with the AICTE / UGC regulations. The academic activities of VFSTR are followed meticulously as specified in the academic calendar as approved by the Academic Council. This academic calendar is shared with all the stake holders well before the beginning of the respective academic year. The curriculum and the course contents of all the programmes are discussed by the respective Board of Studies (BoS), analyzed and recommended for implementation. The Academic Council, being the highest statutory body, chaired by the Vice-Chancellor, meets at least twice or thrice a year and discusses, suggests and approves all the important academic matters related to curriculum and course contents in particular including the recommendations of BoS.

The intended revision in regulations (R22) was in principle accepted and recommended by the Academic Council in its 30th meeting on 07-05-2022. Subsequently respective Board of Studies brought necessary recommendations accordingly, which were duly placed before the Academic Council in its 31st meeting on 30-07-2022.

1.3 Program Duration

For the branch disciplines listed in section (1), the regular courses including theory and practical are offered over a period of four years in eight semesters. The normal duration to complete the B.Tech. program is four years. However, a student can avail the benefit of spill over period for 3 years, that is the maximum duration of seven years can be availed by a candidate to complete the B.Tech. programme in a slower pace if he / she desires. The candidate failing to complete the requirements will be considered for the honorable exit as applicable.

1.4 Courses and Credits

The term course is used in a broader sense to refer to so called papers such as 'theory subject', 'laboratory', 'inter-departmental project', 'major-project' etc. A course can be of theoretical and/ or of practical nature, and certain number of credits are allotted to it depending on the number of hours of instruction per semester. For a course offered in a semester, one hour of lecture (L) instructions carried out in a week is considered equivalent to one credit, whereas two hours of practical (P) sessions done in a week are considered equivalent to one credit respectively. Depending on the course two hours of tutorial (T) sessions may be considered equivalent to one credit. A student earns these credits when he/she successfully completes the course. Add-on credits can also be obtained by successful completion of other recognized co-curricular and extra-curricular activities such as NCC, NSS, Yoga, Dance, Music, Painting, etc. The details of credits of such activities will be provided by the respective course coordinators and assessment of student performance in the activities will be carried out objectively by the constituted committees appointed by the Dean AAA. The criteria of assessment for these activities will include aspects like regular attendance in the programme and satisfactory completion of it through tests conducted at University level or by participation / performance at university level events, state level or national level participation etc. Add-on credits earned in a specified manner will lead to earning B.Tech., B.Tech. with Honours, B.Tech. with Research Honours, B.Tech. with Minor, B.Tech. with Add-on Diploma and B.Tech. with Add-on Certification. Provision is also created for a candidate to migrate from 4-years B.Tech. to 5 years of dual B.Tech. + M.Tech. / MBA degree.

1.4.1 Content Delivery of a Course

Content delivery of a Course in the B.Tech. Degree Program shall be through, either or all, of the following Methods:

i. **Lecture** - refers to Lecture Session(s) through classroom contact session wherein students will learn by listening. Denoted by "L".

ii. **Tutorial** - refers to transaction(s) consisting of Participatory discussion / Self-study / Desk work / Brief presentations by students along with such other novel methods that enable a student to efficiently & effectively absorb and assimilate the contents delivered in the lecture sessions. Denoted by "T".

iii. **Practice** - refers to Practice / Practical sessions and it consists of Hands-on- Experience / Laboratory Experiments / Field Studies / Case Studies / Minor / Major Project, that equip the students to acquire the much required skill component. Denoted by "P".

1.5 B.Tech. Degree

All students formally and conventionally enroll for B.Tech. degree programme. They have to earn **161** credits for the award of degree as specified in the Curriculum. However, additionally he/she can opt to earn up to 20 more credits as Add-on credits, to earn the academic benefits as specified below.

1.5.1 B.Tech. with Honours Specialization: XX

A candidate may earn additional 20 credits, cumulatively totaling to **181** credits in the respective discipline spread over fourth to eight semesters to become eligible for the award of B.Tech. with Honours in YY Engineering (Specialization: XX).

If the add-on credits are designed to take-up a research activity and to complete a research thesis under the scope of 181 credits, then he/she become eligible to receive the award of **B.Tech. with Research Honours in YY Engineering (Specialization: XX)**. However, admission stipulations are applicable for a candidate to pursue B.Tech. with Research Honours. **Annexure-1** provides the supplement regulations for the award of B.Tech. with Research Honours.

1.5.2 B.Tech. with Minor Specialization in XX

A candidate may earn additional 20 credits, cumulatively totaling to 181 credits in anyone of the other disciplines spread over fourth to eight semesters to become eligible for the award of B.Tech. in YY Engineering with Minor Specialization in XX.

1.5.3 B.Tech. with Add-on Diploma

If a candidate earns add-on 20 credits from assorted disciplines/ themes/ centers, then he/ she will be eligible for the award of B.Tech. with Add-on Diploma.

1.5.4 B.Tech. with Add-on Certification

If a candidate in accordance with the sections of 1.5.1, 1.5.2, and 1.5.3 earn additional credits, but will not satisfy the completion of 20 credits, then he / she will be eligible to receive B.Tech. with Add-on Certification. However, such a candidate may also avail a spill over semester to complete 20 credits.

1.5.5 Dual (B.Tech. + M.Tech. / MBA) degree programme of 5-years*

A provision is also created for a candidate who is enrolled for B.Tech. degree to switch over to Dual (B.Tech. + M.Tech. / MBA) degree. **Annexure-2a & 2b** provides the supplement regulations for Dual degree programme.

*Programme will be offered based on satisfactory strength of students willing to register, after receiving the formal AICTE approval.

Note: The consolidated transcript will contain the credits and grade details of all courses amounting to 161 + up to 20 credits.

Every candidate should express his/ her provisional intent at the time of B.Tech. admission for pursuing B.Tech. with Honours/ B.Tech. with Minor/ Dual (B.Tech. + M.Tech. / MBA) degree. However, he / she can exercise his option to change his intent and final confirmation should be submitted at 5th sem. for Research Honours and at 6th sem. for Dual degree.

YY refers to name of the Branch discipline (Eg. CSE / ECE / BT); XX refers to a particular specialization within a Branch discipline.

1.6 Composition of an Academic year

An academic year is composed of an Odd semester (20 – 22 weeks), an Even semester (20 – 22 weeks) and a Summer semester (6 – 8 weeks). The regular semester that begins in July / August is known as odd / first semester and the one that begins in December / January is known as even / second semester (Figure 1). The instructional days for a regular semester shall be a minimum of 90 working days exclusive of days earmarked for summative assessment.

YEAR OF 12 MONTHS											
1	**2**	**3**	**4**	**5**	6	7	8	9	**10**	**11**	**12**
July/ Aug.	Aug./ Sept.	Sept./ Oct.	Oct./ Nov.	Nov./ Dec.	Dec./ Jan.	Jan./ Feb.	Feb./ Mar.	Mar./ Apr.	Apr./ May	May/ June	June/ July
ODD SEM/ FIRST SEM					EVEN SEM/ SECOND SEM					SUMMER SEM	

Figure 1: Distribution of semesters during an Academic Year.

1.6.1 Before the commencement of the semester, a candidate has to pay the stipulated tuition fee and submit an application detailing the courses he / she intended to register, valid for that respective Odd / Even semester. The maximum number of credits per semester will be 25 credits inclusive add-on credits. The intended semester wise coverage will be as presented in the curriculum.

1.6.2 **Summer semester** is a short duration semester program that will be generally conducted during the semester break between even semester and odd semester. The students having 'R' (Repeat grade) courses may register for the course work during this semester to get a chance for successfully completing the 'R' courses. In general, supplementary assessments are conducted in the later part of the summer semester. However, the courses offered in summer semester and the number of courses a student can register are subjected to academic and administrative convenience. A student may register up to a max. of 16 credits in a summer semester.

1.6.3 Exception to the routine practice of registering for 'R' courses in summer semester, a student can register in a course offered by a visiting expert during the summer vacation which may be equivalent to an open elective or a department elective or an Add-on-course. The candidates can register for such courses within the scope of 16 credits. Candidate may also avail summer semester for summer internship opportunities, which may be considered as Add-on credits.

1.6.4 Summer internship credits will also be appended to the credits of Honours / Minor / Add-on certification provided the theme of internship is in accordance with the specialization of Honours/ Minor.

1.7 Semester wise provisions

A student may register for a max of 25 credits per semester as prescribed or otherwise he/ she may include the Repeat courses in the event of having not successfully completed a course or courses in the earlier semester. However, a student may also opt to go in a slower pace to earn the credits less than the prescribed max of 25, including even 'Dropping' a semester for special reasons.

It should be clearly underscored that a candidate should on priority register for Repeat (R) credits if any, during a regular semester, within the said scope of 25 credits; in case he / she cannot be sure of completing the 'R' credits in Summer semester.

1.7.1 During the first four years from the date of admission to B.Tech., a candidate has to pay the semester / annual fees as prescribed irrespective of the less number of credits that he / she would register or even opt to Drop a semester.

1.7.2 If a candidate gets into spillover semester beyond four years up to a maximum of seven years he / she has to pay semester fee proportional to the credits that he/ she registered in that spill over semester as prescribed from time to time.

1.7.3 A candidate has to pay additional fee proportional to the number of credits for registering in a summer semester as prescribed from time to time.

2. CURRICULUM

Each School offers different B.Tech. degree programmes and the departments concerned prescribes semester-wise curriculum encompassing different courses. Every course offered will be designated in a L-T-P structure. The theory courses comprise of L (and / or T & P hours) whereas the practical courses include instructions (T) and practical sessions (P). Amalgamation of theory courses with practical sessions is predominantly seen in this curriculum. Specifically, during the first year, some of the courses may be offered in a swap mode, that is the same course is offered in both the semesters of I-year to accommodate the student strength and for operational convenience.

2.1 Distribution of credits

The overall distribution of credits for various categories of courses in the curriculum of B.Tech. programmes is represented in Table (1) as given below.

Table 1 : Credits Distribution for Various categories of courses.

Category of Courses	Number of Credits	Percentage of Credits	AICTE Recommendation (%)
Professional Core	53	32.92%	30 - 36%
Electives (Department & Open / I^2C)	34	21.12%	19 - 23%
Basic Engineering	20	12.42%	10 - 18%
Humanities and Management	15	09.32%	08 - 09%
Basic Sciences	19	11.80%	12 - 16%
Life Skills / Physical Fitness / Floating Credits	06	03.72%	--
Projects	14	08.70%	08 - 11%
Total	**161**	**100%**	

2.2 Organization of course contents

Courses offered in the program is composed of two modules covering all the course contents required for a candidate to obtain knowledge and skill. Content in each module is further distributed among two units; wherein Unit -1 contains 'Fundamentals and Broad perceptive' of the module. Unit-2 comprises of the extension / advanced topics of Unit-1 as well as necessary practice models for validation / applying the knowledge gained during L/T sessions. The modular period is about 8 weeks. The first unit in a module may be covered in 2 to 3 weeks and the second unit of the module maybe of 5 to 6 weeks (Figure 2). By the end of each module a candidate must be in a position to translate his/ her L-based knowledge into P-based skill as prescribed in the curriculum. Individual formative assessment shall be in place for each module and a single semester-end summative assessment for the course composed of both the modules.

YEAR OF 12 MONTHS											
1	**2**	**3**	**4**	**5**	6	7	8	9	**10**	**11**	**12**
July/ Aug.	Aug./ Sept.	Sept./ Oct.	Oct./ Nov.	Nov./ Dec.	Dec./ Jan.	Jan./ Feb.	Feb./ Mar.	Mar./ Apr.	Apr./ May	May/ June	June/ July
ODD SEM/ FIRST SEM					EVEN SEM/ SECOND SEM					SUMMER SEM	

Module- I		**Module- II**			**Module- I**		**Module- II**			
U1	**U2**	U1	U2		U1	U2	U1	U2		

Figure 2: Unit-wise distribution of course contents in a module and their mapping with Academic Calendar; U= Unit.

2.3 Prerequisite Knowledge

Wherever prerequisite knowledge was mentioned it should be treated as registering for the courses covering the content mentioned in prerequisite knowledge are mandatory for the student before he / she register in a higher level course.

3. CHOICE BASED CREDIT SYSTEM

Each branch discipline of the B.Tech. programme comprises of a set of courses - basic sciences, humanities and management, basic engineering, professional core, electives, employability & life skill courses. VFSTR offers flexibility for students to choose courses of their choice and obtain the credits satisfying the minimum credits criterion in each category as given in Table (1).

3.1 Common (Core) Courses for all branch disciplines

Majority of the basic science courses, basic engineering courses, humanities and management courses and general life skill courses including a course on orientation are offered commonly to students of all disciplines of study.

3.1.1 Basic Sciences and Humanities

Basic science courses are included to offer the knowledge of scientific theories that form the foundation for all the engineering solutions. Courses of Management and Humanities are also offered. The main purpose of offering Management courses is to impart Management skills to students so that they would pursue allied career opportunities. 'Humanities' courses are meant for gaining managerial and organisational skills. In this some aspects of civil services are also covered for the benefit of those interested students.

Orientation course : Course content covers cross-cutting issues of the society like Gender Equality, Human Values, Professional Ethics, Moral and Ethical Values. Orientation course is offered as Binary graded course for I year B.Tech. students and on successful completion earns 3 credits. All the students have to register during first / second semester and mandatorily obtain successful completion grade.

3.1.2 Basic Engineering

Basic engineering courses are those that every engineer, irrespective of his discipline, ought to understand to perform well in his/her discipline. Engineering drawing, IT workshop, computer programming are some of the courses that fall in this category.

3.2 Professional Core

Professional Core courses are individualized for each programme and they are mandatory for every student opting for that branch discipline. These are designed to offer the essential fundamental knowledge and skills required for that specific programme.

3.3 Electives

A candidate has a choice to choose the elective courses. A list of elective courses is pooled together, enabling a candidate to choose the electives from a pool so that he/she can focus to a specific theme. Otherwise also he/ she can exercise the choice to choose electives from across the pools. There may be courses which may not be listed under any pool, which are called 'Free elective courses'.

Elective courses offered for each programme are categorized as 'department electives' and 'open electives'. Some electives may have pre-requisites *i.e,* successful completion of a course / knowledge of a course / the department's consent.

3.3.1 Department Electives

'Department electives' are those courses that are aimed at offering the advanced/ additional knowledge in the chosen branch discipline.

Care should be exercised while opting for departmental elective courses and specialization courses that is a course opted as departmental elective should not be registered as a course under Specialization and vice versa.

3.3.2 Industry – Interface (I^2) courses

I^2 courses are offered with the support of industry personnel and are also included in department elective category. These courses are of one credit, and are offered by experts from industry / academia. The I^2 course is offered to 3rd year B.Tech. students with 15 hours' duration as Binary graded course. The duration of 15 hours can be spread across 3 days continuously or 3 different slots of 5 hours each on consecutive weeks. Students who successfully complete the course will be awarded one credit.

3.3.3 Open Electives

The elective courses offered by the departments/ centers/ themes for other department students are called as open electives. These electives are offered to enable students understand interdisciplinary nature of courses. A student's enrollment for pursuing an open elective course(s) is based on his/her choice and order of merit and subject to availability of seats.

Elective courses are spread over four semesters from fourth to seventh semester, to enable students earn credits from a chosen pool or otherwise.

Care should be exercised by the candidate while opting for Open electives and minor courses, that is a course opted as Open elective should not be registered as a course under Minor and vice versa.

In 'electives' category, a student is required to secure a minimum of 34 credits. These credits can be obtained through a combination of courses from his / her own department and other departments. Of these 6 credits have to be earned through MOOCs offered via Swayam platform. A student may also be permitted to earn more elective credits through MOOCS. This will promote self-learning and drive students towards innovative learning approaches.

To facilitate the process of acquiring the elective credits through MOOCS, VFSTR has constituted "Online course committee" at Central level with 1-2 members represented from each of the department to guide students in selection of courses and to assist them with further steps if required until completion of the courses.

3.4 Inter-departmental Project

These projects are designed and executed by students during the third year of their program. By doing these projects, students will get an idea of how technologies or processes, prototype or working model can be developed by culmination of technologies from courses of different programs. The supervision of inter-departmental projects will be done by faculty from different departments who serve as supervisors. The minimum duration of inter-departmental project during each of the semester is 90 hours including writing of project report and submission for assessment. A batch of 2-5 students from across the departments can take part in each of the inter-departmental projects. Performance will also be assessed in the modular framework for formative and semester-end summative, however spread over 2 semesters with a weightage of 2 credits.

3.5 Project

Students may opt for Project work in lieu of internship. Such students may avail research -internship support from any institution well known for research and development (R&D). They may also take up project work in VFSTR itself. Each candidate has to submit interim reports and a final report which are mandatory requirements towards the partial fulfillment of project credits requirements. It bears a weightage of 12 credits with a duration of 90 working days. During the semester the student under the guidance of a faculty member(s) will involve in an innovative design / research through the application of his / her knowledge gained in various courses studied. He / she is therefore expected to present a survey of literature on the topic, work out a project plan and carry it out through experimentation / modelling / simulation / computation. Through such a project work, the student is expected to demonstrate system analysis, design, presentation and execution skills. Performance in the project will also be assessed in the modular framework for formative and semester-end summative.

3.6 Internship

A student can undertake internship in lieu of project work in industry for one complete semester during seventh/ eighth semester in lieu of major project work. If the Internship is under taken during 7th semester, then the regular course work of 7th semester should be taken up during the 8th semester (7th and 8th semester happen in a swapped mode). It bears a weightage of 12 credits. This is aimed at training students in solving / understanding real-life problems through application of engineering analysis, design, evaluation and creation, particularly in association with practitioners and experts in the industry. The procedures for obtaining the internship placements and allocation of the same to the students are as per University defined norms outlined in the 'internship programme operational guidelines' manual. Even during internship, a student is preferably expected to carry out a focused study on one topic/ problem in consultation with the interning institute. Internship progress report should be submitted periodically and finally a detailed internship report should be submitted duly certified by a mentor from the internship institute. Performance in the internship will also be assessed in the modular framework for formative and semester-end summative.

3.7 Binary Graded courses

Apart from orientation course, I^2 courses, following courses shall also be offered as Binary Graded courses.

3.7.1 Physical Fitness

As physical fitness contributes to physical, mental and social development, it is offered to students. The credits are spread across 2 semesters in I year with 1 credit per semester. Students are imparted training through physical exercises.

3.7.2 Life Skills

Realizing the VFSTR's Vision of preparing multifaceted personalities, VFSTR floats various credit based life skill activities for students such as yoga, dance, music etc. The credits are spread across 2 semesters in II year with 1 credit per semester.

3.7.3 Courses with floating credits

With a view to imbibe social responsibility and all round development students are encouraged to participate in NCC / NSS / Students Activity Council (SAC) / Entrepreneurship cell (E-cell) / Student Mentoring / Social activities / Publication with good impact factor (Only 2 students can claim 1 paper / patent). Such activities lead to awarding of 2 credits.

In case the candidate does not successfully complete the binary graded courses, they will be placed under 'R' category. However, such R should be completed in regular Odd / Even semester whenever these courses are offered. In special cases provision of summer semester may be provided.

4. ATTENDANCE

It is mandatory for the student to attend the course work in each semester as per the academic schedule of that semester. VFSTR expects 100% attendance. However, the attendance in each course shall not be less than 75 % of the aggregate of all L, T, P sessions conducted in that course.

a) The attendance calculations will be periodically reviewed at the end of every 4 weeks. The details of attendance status will be shared with the parents / guardian. The final status of attendance will be reported at end of 15th week granting the advantage of the attendance for the 16th week for the purpose of attendance shortage calculations.

b) The shortage of attendance may be condoned up to 10% on the ground of ill-health, social obligations, participating / representing in sports/cultural events, placement activities etc.

c) Documentary evidence like medical reports and certificates issued by concerned bodies is to be produced on time as support for the attendance shortage due to ill-health. These cases are subjected to the scrutiny of a committee constituted for this purpose by the Vice-Chancellor. The decision of the committee shall be final.

d) Prior approval has to be taken from the HoDs for the other types of leaves.

e) The courses where the student shortage of attendance was not condoned shall be considered as 'Repeat' category courses and will be under 'R' grade in the student's semester transcript. Student should re-register for these courses during the summer semester or whenever the course is offered next time during regular semesters. These re-registrations are subjected to the regulations at the time of re-registration. In case of core courses, the same core has got to be re-registered. However, in case of an elective a candidate may exercise a choice of choosing different elective in place of 'R' graded elective.

The students who are put into 'R' grade will not be allowed to take up the summative assessment in that semester. In case due to lack and/or delay in information, if he/she appears for the summative assessment in that course, office of AAA is empowered to cancel the attended exams. The scores obtained either in formative or summative assessment will not be considered for grading.

5. ASSESSMENT

Teaching-Learning and Assessment should go hand in hand and complement each other. Continuous assessment plays a vital role to enable the student to get synchronized with the teaching-learning process. Assessment mechanism adopted in the institute is aimed at testing the learning outcomes in tune with the outcome based model of education. The focus, is thus on assessing whether the outcomes are realized by the end of the course.

The performance of a student in each course is assessed on a continuous basis during the semester through various in-semester and end-semester assessment models. The marks awarded through continuous assessment are referred to as Formative assessment marks. The marks awarded through end-semester tests are referred to as Summative assessment marks (Figure 3). Both the formative and summative assessment marks are considered for awarding the final marks and the grade point in a particular course.

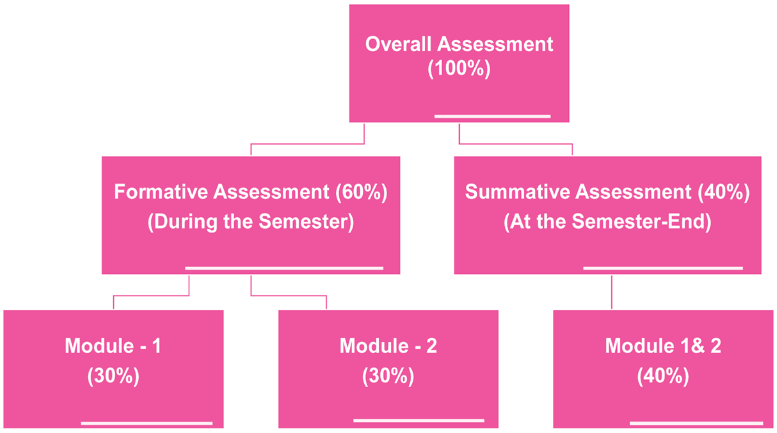

Overall Assessment (100%)
Formative Assessment (60%) (During the Semester)
Summative Assessment (40%) (At the Semester-End)
Module - 1 (30%)
Module - 2 (30%)
Module 1& 2 (40%)

Figure 3: Categories of assessments in place for R22.

5.1 Marks distribution

For each course, the maximum sum of formative and summative assessment marks put together is 100, in the ratio of 60:40, respectively.

5.2 Qualifying criteria

To be declared successful in a course, a student must secure at least a grade 4.0 in a scale of 10 based on the total maximum marks which is inclusive of formative and summative assessment. The students should also get 35% from the maximum marks allotted for formative and summative assessments individually.

The hierarchy of qualifying criteria is as follows:

i. Attendance compliance should be 75% or within condonable range; else the candidate is put into 'R' grade.

ii. In formative assessment, a candidate should secure a minimum of 35% *i.e*. 21 marks out of 60; else the candidate is put into 'R' grade.

iii. In summative assessment, a candidate should secure a minimum of 35% *i.e.* 14 marks out of 40; else the candidate is put into 'I' (Incomplete) grade.

iv. Collectively the candidate should secure a min. grade of 4.0 in a scale of 10 after relative grading (section 7); else the candidate has to choose either 'R' or 'I' grade duly being counselled.

v. A candidate who has secured grade <5 in a course may be permitted (optional) to volunteer to improve his / her grade by opting suitably 'R' or 'I' grade in that course.

The candidates with 'R' grade should re-register for 'R' courses either in Summer semester or in a regular semester as and when the courses are offered. The candidates in 'I' grade are allowed to appear for supplementary summative assessment whenever the semester-end assessments are conducted.

To assess Binary graded courses / special projects / courses, not fitting into the categories described here, a suitable assessment procedure will be evolved in consultation with experts of that area and adjudicated by the committee constituted for that purpose. The decision given by the committee will be final. The appended assessment scheme shall be announced by the course coordinator during the commencement of course.

YEAR OF 12 MONTHS											
1	**2**	**3**	**4**	**5**	6	7	8	9	**10**	**11**	**12**
July/ Aug.	Aug./ Sept.	Sept./ Oct.	Oct./ Nov.	Nov./ Dec.	Dec./ Jan.	Jan./ Feb.	Feb./ Mar.	Mar./ Apr.	Apr./ May	May/ June	June/ July
ODD SEM/ FIRST SEM					EVEN SEM/ SECOND SEM					SUMMER SEM	

Module- I		**Module- II**			**Module- I**		**Module- II**			
U1	**U2**	**U1**	**U2**		**U1**	**U2**	**U1**	**U2**		
Formative Assessment				SA	Formative Assessment				SA	

Figure 4: Schedules of formative and summative assessments in line with Academic calendar. SA = Summative assessment.

5.3 L-based courses integrated with P/T

5.3.1 Formative Assessment

The scheme of formative assessment is designed to promote the continuous learning. Scheme consists of assessments planned at institute level and assessment that may be scheduled by the course instructor (Figure 4). Institute level assessments shall be scheduled by the office of AAA. Respective Faculty Member(s) shall declare the schedule of Continuous Laboratory Assessments (CLA), Quiz, Tutorials, Assignments, Seminars, Discussions, etc. Some of the components may also however take place in an unscheduled manner like Surprise Tests. However, students shall be made aware of the assessment modalities that are going to be followed in a course by the faculty, under information to the HoD.

To monitor the progress of students, continuous assessment comprising of five targets (T1, T2, T3, T4 and T5) is advocated in each module for a maximum of 60 marks. For a class (or section) of 60 to 70 students, formative assessment commences by the announcement of module bank containing 10 problems for each module in a course. Nature of problems in the module bank shall be at the level of creative / exploratory / design / thought provoking covering the complete syllabus of a module at somewhat advanced / challenging level.

The purpose of creating module bank of 10 problems is to assign one problem each to 2 batches of 3 - 4 members. The batches are composed of randomly picked up candidates. These batches remain same for all courses and also for the P-sessions in the courses in that semester and are created in the beginning of that semester.

The purpose of assigning one problem to two batches is to create a healthy competitive spirit between the two batches.

The modality of evaluation of five targets is listed here under:

a) **T1**: During 5th or 6th week of each module a classroom test shall be conducted. T1 consists of two parts: A and B.

Part A consists of one random problem from the module bank and vary from batch to batch. All the questions in the module bank shall be distributed among students and students shall know the question to be answered only on the day of test in the examination hall.

Part B consists of one common problem at fairly application/ advanced level (**not at all prior notified**) from outside the module bank for all the students.

T1 shall be paper based and proctored test for a period of 60 min (maximum) which shall be assessed for 30 marks and downscaled to 10 marks.

For the students who for justifiable reasons could not attend the classroom test on the scheduled day, a re-test maybe conducted. However, Part-B will contain a new question and Part-B will have higher weightage than part-A or full weightage could even be allotted for Part-B in such an event.

b) **T2**: Immediately follows T1. Students in a specified batch who now have received the same question during T1 will work further on that problem for T2.

T2 is primarily an extension of problem received in T1 for carrying out validation study: Case studies / Simulations / Experimentation. Each batch shall interact with the course instructor to finalize the nature of validation and expected to complete the exercise within 10 to 15 days after T1.

Course instructor should ensure assigning a different case study / a different scope for validation study for each batch in case the same problem is assigned to two batches.

Course instructor shall assess every student in a batch for a max. of 10 marks based on his observation, interaction and/or reviewing (based on at least two reviews).

c) **T3**: T3 shall be conducted during the last week of each module. Student batches are expected to submit a report, clearly documenting the work executed during T2. The report should be in IEEE / APA format and additionally a voice in-built PPT should be prepared and submitted.

The report and presentation shall be assessed by the course instructor for 10 marks for every student. In certain cases, a course instructor can call for a physical presentation also by a batch.

d) **T4**: T4 is a comprehensive module test, conducted for 30 min. comprising of 20 multiple choice questions (MCQs) covering the holistic content of module. T4 shall be evaluated for a max. of 10 marks @ ½ mark for each question. T4 will be conducted in ON-LINE mode.

There shall be two tests in each course in a day and the best performance of the tests shall be considered for awarding the marks.

Two sets of question papers each containing 20 questions should be set. The theme of the questions could be similar across the sets. When the test is administered online, every student receives the questions in shuffled sequence and also the choices in shuffled sequence. Therefore, the choice like both ‘a’ & ‘b’ above. Neither ‘a’ nor ‘b’, all the three a, b, c will not be set.

e) **T5:** T5 assessment is based on Practice or Tutorial assignments. Implementation, Report presentation and Discussion shall happen in a continuous mode throughout the module period.

At least 4 such continuous lab practice assessments (CLPA) / assignments per module shall be conducted by course instructor. The marks will be @ 5 marks per assignment totaling up to 20 per module.

f) The scores of the targets are to be normally announced within three working days on completion of the assessment and the performance is to be discussed in the class.

g) The total marks per module is 60 - T1 (out of 10), T2 (out of 10), T3 (out of 10), T4 (out of 10) and T5 (out of 20).

h) Total marks for both the modules from formative assessment will be added up to 120, which will be ***suitably mapped down*** to a max. of 60 marks. The mapping policy should be decided by the lead instructor / instructors in consultation with the HoD. The mapping policy should be shared with Dean AAA for the purpose of documentation.

i) The marks scored in Module-1 for a max. of 60 should be entered / submitted latest by 9th week and of Module-2 latest by 17th week of the semester. Consolidated score of for a max. of 120 ***suitably mapped down*** to a max. of 60 marks should be submitted latest by 18th week of semester enabling the declaration of 'R'- grade before the commencement of summative assessment.

j) A candidate put under 'R' will not be permitted to take up the summative assessment.

5.3.2 Summative Assessment

a) An instructor may choose one of the two formats for conducting summative assessment for L-based courses integrated with T/P.

i) 15 + 25 marks format or 20 + 20 marks format (following b, c, d below).

ii) 40 marks format (following c, d below).

b) **If summative assessment is in two parts format:**

i) Part-I will be the assessment of capstone project which is pre-assigned during the module-2 period or will be the exploratory review assessment of all lab practice assignments.

ii) Part-II will be based on a written examination for a max. marks of 80, as in c & d below, which is ***scaled down*** to 25 or 20 based on the selected pattern of format.

iii) A candidate should attend both the parts of summative assessments; else he will be put into I grade.

c) For each L-based course integrated with T/P, the summative assessment shall be conducted by the Institute for a duration of 150 min. and for a maximum of 80 marks. Contents for summative assessment shall cover the breadth and depth of the complete syllabus that is mentioned in the two modules of a course.

d) The question paper for end-semester theory examination consists of two parts as given in Table (2).

Table 2: L-based Summative Assessment Question Paper Pattern.

Part No.	No. of Questions	Marks for each Question	Marks	Choice
A	4	10	40	No
B	2	20	40	No
Total Marks			80	

e) The questions will be comprehensive covering the entire course syllabus and any single question should not necessarily be limited to any particular unit / module.

f) These marks are ***suitably mapped down*** to a score of 40.

g) Total marks of summative assessment will be for a max. of 40 irrespective of format of evaluation.

h) The award of 'I' grade is solely based on marks scored in summative assessment out of 40, if he/she does not score a min. 14 out of 40 (35%).

5.4 P-based Courses

The detailed information consisting of experiments, batch formations, experiment schedules, etc., will be displayed / informed to the student in the first week of the semester so that the student comes prepared for the lab sessions. Copies of the lab manual will be made available to the students along with the schedule. The lab manual will consist of the list of equipments, detailed procedure to conduct the experiment, format for record writing, outcomes for each experiment and possible set of short questions to help students gain critical understanding. The courses like workshop (IT), Engineering Graphics will also be treated as P-based courses.

5.4.1 Formative Assessment

During laboratory sessions, a brief viva-voce is conducted for each student on the experiment he/she is carrying out on that day. Some of the parameters that could be included in the Continuous Lab Assessment (CLA) are given in Table (3). The set of parameters may slightly differ from one laboratory to the other, and will be announced before the commencement of the lab session. These parameters are assessed for each laboratory session.

Table 3: Suggested parameters for Continuous Lab Assessment (CLA).

S. No	Component	Marks
1	Report of about 1 page on proposed experimental layout and background theory before the start of lab session	4
2	Viva and interaction to evaluate understanding of concepts	4
3	Experimentation and data collection	4
4	Analysis of experimental data and interpretation	4
5	Finalized report submitted in the next week	4
Total		**20**

This assessment is carried out for each practical session and the total marks of all practical sessions will be ***suitably mapped down*** to a max. of 60.

5.4.2 Summative Assessment

End semester examination for each practical course is conducted jointly by two examiners. The examiners are appointed by Dean, AAA from the panel of examiners suggested by the respective Heads of the Department. In some cases, one of the examiner may be from outside the institution and will be identified as external examiner. The scheme of assessment may vary depending on the nature of laboratory, which shall be shared with student by the laboratory in-charge. The summative assessment will be conducted for a max. marks of 40. The general scheme of assessment is given in Table (4).

Table 4: Suggested end-semester summative assessment pattern for P-based courses.

Component	**Marks**		
	Examiner 1	**Examiner 2**	**Total**
Objective & Procedure write up including outcomes	4	4	08
Experimentation and data collection	4	4	08
Computation of results	4	4	08
Analysis of results and Interpretation	4	4	08
Viva Voce	0	8	08
Total Marks	16	24	40

5.5 Assessment and Grading of MOOCs based elective

Whenever a candidate opts for a course through MOOCS offered via Swayam platform, he / she has to learn and undergo assessment as per norms set by VFSTR for such MOOCs Courses. Upon the declaration of the result, that the candidate has successfully completed the course, the candidate is said to have earned the credits under credit equivalence and credit transfer.

5.6 Inter-departmental Project

Inter departmental project work is undertaken in the 5th semester and continued till the end of 6th semester for earning 2 credits by each candidate. It is expected that the inter-departmental Projects result in publication of a technical paper in a peer-reviewed journal. For this purpose, Dean R&D and Heads of Department will identify faculty mentors who will guide the students by conducting workshops on drafting of research article, communication, process of peer-review, publication, etc. The faculty will support the students by all means to get the technical findings published in peer-reviewed journals.

5.6.1 Formative Assessment

The assessment will be carried in four reviews in a systematic way wherein; first two reviews are conducted in the 5th semester and remaining two reviews in the 6th semester. Reviews will be conducted jointly by the two supervisors allocated by the two departments representing the host and the collaborating department.

The detailed assessment guidelines and scheme are to be announced along with the assessment schedule as mentioned in the Table (5).

Table 5: Schedule and parameters followed for formative assessment.

Semester	No of Module	Schedule	No of reviews	Points to be considered	Formative assessment marks
5th semester	Module -1	7th – 8th week	Review -1	• Identification of specific area out of broad areas under the supervisor • Identification of outcomes in line with programme objectives. • Feasibility of contributing to the attainment of outcomes • Identification of tools / equipment / surveys / training needs / etc..	15
	Module - 2	15th – 16th week	Review -2	• Understanding by individual students on the overall aspect of the project. • Completion of literature survey • Design of equipment set up • Acquisition/learning of the tool required • Readiness of about 25% documentation	15

6th semester	Module - 1	7th - 8th week	Review -3	• Progress review as per mechanism/ schedule identified • Evaluation or validation techniques of the project • Structure of project report • Individual student contribution in above activities	15
	Module - 2	15th-16th week	Review -4	• Presentation of results, analyses and conclusions • Meeting of objectives defined in first review • Preparation of report • Understanding by individual students on the overall project • Submission of technical article	15

5.6.2 Summative Assessment

Summative assessment will be done jointly by two examiners one is from VFSTR as internal examiner and the other may be from other institution as external examiner. These examiners will be appointed by Dean-AAA from the panel of examiners suggested by the respective Head of the Department. The scheme of assessment will be report (15 marks), presentation (10 marks) and demonstration (15 marks) respectively. Points to be considered during the review.

a) Presentation of results, analyses and conclusions

b) Meeting of objectives defined in first review

c) Preparation of report

d) Understanding by individual students on the overall project

e) Individual student contribution

f) Progress of project as per schedule

g) Submission of technical article/ prototype realization

5.7 Project

Those students who do not opt for the semester-long internship, carry out their major project at VFSTR and submit their report which is a mandatory requirement for the award of degree. These projects are usually done in batches (not exceeding five students in a batch), during the VIII semester, under the guidance of a faculty member. Every batch, in consultation with the guide, should define the project and also the probable procedure of carrying it out and submit the same to a committee consisting of 2 to 3 faculty members appointed by Head of the Department. This is to avoid the repetition and also to come up with a roadmap for completion of the project within the time stipulated. The students are encouraged to select topics related to ongoing research and consultancy projects. The students are expected to carry out and present a survey of literature on the topic, work out a project plan and its implementation through experimentation / modelling / simulation / computation. They are also expected to exhibit system analysis, design, and presentation and evaluation skills. The entire process of grouping of student batches, and identification of respective guides etc., is to be completed by the end of VII semester, so that students can start of their project work immediately after VII semester.

5.7.1 Formative Assessment

The progress of project is reviewed twice in a module by the Project Review Committee (PRC) and formative assessment marks are awarded based on these reviews. The Project review committee consists of

a) Head of Department or his/her nominee – Chairperson

b) A senior faculty member identified by the HoD – member

c) Project supervisor – member

Review schedules of PRC are to be announced by the department immediately after the commencement of class work. The review presentations are open to all the students of that section and attendance is compulsory. The first review should be of 15 minutes / batch; the remaining reviews should be around 30 minutes / batch. Before every review the batches should submit their PPT along with a brief report of not exceeding two pages. It is to be expected by the committee that student communicates/publishes research article based on the project work prior to graduation. The following aspects may be considered by the committee for assessment Table (6).

Table 6: Schedule and suggested parameters to be considered for formative assessment.

Module	Schedule	Review	Points to be considered	Max. Marks
Module -1	4th week	First review	● Identification of specific area out of broad areas. ● Identification of outcomes in line with programme objectives ● Feasibility of contributing to the attainment of outcomes	15
	8th week	Second review	● Identification of tools/equipment/training needs/ etc.. ● Understanding by individual students on the overall aspect of the project ● Completion of literature survey ● Design of project set up	15
Module – 2	12th week	Third review	● Acquisition/learning of the tool required ● Readiness of the layout of the project report ● Progress review as per mechanism/ schedule identified ● Individual student contribution in above activities	15
	16th week	Fourth review	● Presentation of results and conclusions ● Meeting of objectives defined in first review ● Submission of draft report ● Understanding by individual students on the overall project ● Individual student contribution ● Progress of project as per schedule	15

5.7.2 Summative Assessment

At the end of the semester, during 18th to 20th week of the semester the summative assessment will be conducted in two phases.

Phase–I (during 18th -19th week): This is an evaluation for a max of 20 marks. A committee of two members comprising of HoD's nominee and Guide will assess the project work which will involve going through the project report (6 marks), project presentation (7 marks) and demonstration of the project (7 marks).

Phase–II (during 20th week): A final presentation and defense assessment for a max. of 20 marks will be carried out by one-man committee composed of an external expert who is chosen by the Dean AAA from a panel of examiners suggested by the HoD. The format for evaluation will involve going through the project report's quality (6 marks), presentation (6 marks) and interaction and defense (8 marks).

The qualifying marks will be finalized considering the marks scored in both the phases (I & II) of summative assessment.

In case the candidate is placed in 'I' grade, he/she has to appear for both Phase-I and Phase-II assessments, which will be held within the 15 days after declaration of results. In the consecutive assessment also if the candidate fails to secure min. required score then he/ she will be placed in 'R' grade.

5.8 Semester-long Internship

Internship work is undertaken either in the VII or VIII semester by a student in an industry, under the joint supervision of industry personnel and an internal faculty member. Sixty percentage of the marks of Internship are allotted through continuous evaluation as formative assessment and the remaining 40% are based on end semester assessment.

Table 7: Assessment scheme for Internship reviews.

Module	Review	Schedule	Formative assessment marks
Module -1	First review	4th week	10
	Second review	8th week	15
Module – 2	Third review	12th week	15
	Fourth review	16th week	20
Total			**60**

a) The progress of internship work is reviewed twice in every module by the "Internship Review-Committee" and marks for formative assessment are awarded based on these reviews.

b) The Internship Review Committee (IRC) consists of Head of Department or his/her nominee (Chairperson), the internal and external (industry) supervisors.

c) The IRC may not be the same for all students; however, the same IRC should exist for entire duration of the internship program of any single student.

d) The schedule and the scheme of evaluation are to be announced with internship notification. The internship reviews may take place at the place of internship or at the university, as decided by the interning organization or may be conducted in the blended mode.

5.8.1 Formative assessment: Internal reviews at the place of internship

The internal supervisor will interact with the guide allotted at internship offering industry based on the schedule given to conduct the reviews. Scheduled reviews can be conducted by IRC on online mode for discussion/ presentation. The 15 marks obtained by students for each review will be scaled to allotted marks as given in Table (7).

a) Students should submit a report (not more than two pages) explaining about the progress of their work, mentioning clearly details like the machines or software handled / adopted, type of data collected and his/her understanding and contribution in the programme, and the same has to be presented before the supervisors.

b) The candidate should clearly present the completion of stipulated assignments set by the industry supervisor for that period.

c) The evaluation will be based on a & b above and also based on regularity and discipline maintained in the internship venue.

Table 8: Suggested scheme of assessment for every review

Component	Total
Regularity and interaction	3
Application of knowledge	2
Gaining of new knowledge /skills / literature survey	5
Internship progress	3
Report	2
Total marks	**15**

5.8.2 Summative assessment – Internship

At the end of the semester, the student shall submit a comprehensive report of internship covering the work done and make a final presentation in two phases as follows:

Phase–I (during 18th -19th week): A committee of two members comprising of internal supervisor and HoD's nominee will assess the overall internship participation by the candidate and his final report through presentation made by the intern. The internship report (6 marks), presentation (7 marks) and overall impression (7 marks) during the internship will be evaluated respectively.

Phase–II (during 20th week): A final presentation and defense assessment for a max. of 20 marks will be carried out by one-man committee composed of an external expert who is chosen by the Dean AAA from a panel of examiners suggested by the HoD. The format for evaluation will involve going through the project report's quality (6 marks), presentation (6 marks) and interaction and defense (8 marks).

The qualifying marks will be finalized considering the marks scored in both the phases (I & II) of summative assessment.

In case the candidate is placed in 'I' grade, he / she has to appear for both Phase-I and Phase-II assessments, which will be held within the 15 days after declaration of results. In the consecutive assessment also if the candidate fails to secure min. required score then he/ she will be placed in 'R' grade.

6. SEMESTER-END ASSESSMENT ACTIVITIES

6.1 Setting of semester-end summative assessment question papers will be coordinated by the lead instructor assigned for a particular course. Two sets of question papers will be submitted latest by 12th week of the semester.

6.2 There shall be 'Summative Assessment Question Paper Scrutiny Committee' which would be constituted with external experts. Experts are empowered to modify / rephrase the questions to maintain a high standard of the semester-end assessment. The review should be completed by the 14th week of the semester. The review process will be coordinated by a committee of School Dean, HoDs and external experts.

6.3 The question wise marks scored in the summative assessment out of a total of 80 will be made available online within two weeks from the last date of examination and would be kept active for 24 hours. Latest by the end of 48 hours from the instant of notification any candidate can submit an appeal online providing question wise claim.

6.4 Claims for re-assessment on P-based courses are not allowed.

6.5 The appeals will be attended within next three working days. Fees for appeal, as decided from time to time, has to be remitted online along with the appeal.

6.6 Final results and grades will be computed as explained in the next section.

6.7 Final results and grades shall be announced within four weeks of completion of the last examination of the summative assessment (within two weeks from the last date of appeal). Grades are published on the University website, and also informed to the parents and students through SMS.

6.8 Provisional Grade cards will be issued within two weeks after the announcement of grades. Grade card will contain three parts. Part 1: Details of successfully completed courses. Part 2: Details of 'I' grade courses. Part 3: Details of 'R' grade courses.

7 COMPUTATION OF GRADING

7.1 Formative assessment decides the list of 'R'- candidates. Therefore, these candidates will not be considered for grading computation. Summative assessments decide the list of 'I' candidates. Therefore, these candidates will not be considered for grading computation.

7.2 The candidates who have successfully completed both formative and summative assessments will be considered for computation of relative grading.

7.3 Threshold value (**Th**) for relative grading in each course is arrived after studying the marks distribution in that course by a committee constituted by office of Dean AAA. The threshold value is decided by the upper bound marks of the major chunk of the class keeping the top outlier scores away from consideration (the least upper bound). The threshold value will be slightly greater than upper bound marks or may be equal to the upper bound marks.

7.4 The total marks (**m**) = marks scored in the formative assessment + marks scored in the summative assessment is transformed into relative grade expressed accurate to two decimal places as follows:

Relative grade point (P) = (m/Th) X 10 [and limited to 10]

7.5 If students require course wise percentage equivalence, then the calculation will be based on the following

Course wise percentage equivalence = (m/Th) X100
[truncated to two-digit integer and limited to 100]

7.6 After relative grading, a student is assigned a 'Letter Grade (G)' for each course as per Table (9). The grade and the corresponding letter grade represent the outcomes and assessments of a student's performance in a course.

Table 9: Grading information

Relative Grading Range (P)	Category	Grade (G)
≥ 9.50	Outstanding	O
≥ 8.50 to 9.49	Excellent	S
≥ 7.00 to 8.49	Very good	A
≥ 6.00 to 6.99	Good	B
≥ 5.00 to 5.99	Fair	C
≥ 4.00 to 4.99	Marginal	M
Transitional Grade	Repeat	R
Transitional Grade	Incomplete	I

8. SUPPLEMENTARY EXAMINATIONS

8.1 The supplementary examinations shall be conducted once in summer semester. Notifications will be released by the examination section informing the students about registration procedures, details of fee and timetables. Apart from these examinations the students who have courses with 'I'-grade can also write the supplementary examinations along with regular semester-end examinations of that academic (Odd / Even) semester.

8.2 Whenever a candidate clears courses with 'I' grade in a supplementary examination that are conducted during a regular semester, the Threshold value for computing his / her grade will be obtained from the same batch in which he / she had completed his/her formative assessment.

8.3 Whenever a candidate clears courses with 'R' / 'I' grade in a summer semester, the Threshold value for computing his / her grade will be carry forwarded from the preceding Odd / Even semester for the respective courses.

8.4 Whenever a candidate clears courses with a 'R' grade in a regular semester along with his/ her junior batch then for this candidate the Threshold value will be corresponding to his/her junior batch for computing grade.

8.5 The results of summative assessment of Project / Internship will be announced only if the candidate successfully earn all the credits in courses registered during the program. If the candidate is with 'R' / 'I' graded courses the results will be kept under **'Announced Later (L)'** status and will be announced only after candidate clears these courses.

9. GRADE POINT AVERAGE

The Academic Performance of a student in every semester is indicated by the Semester Grade Point Average (SGPA) and finally by Cumulative Grade Point Average (CGPA).

9.1 SGPA

The Semester Grade Point Average (SGPA) shall be computed using the formula given below:

$$SGPA = \frac{\sum_{i=1}^{n} C_i P_i}{\sum_{i=1}^{n} C_i}$$

Where

n = number of courses a student successfully completed in the semester under consideration

P_i = Grade points secured for the i^{th} course registered in the semester under consideration.

C_i = the number of credits assigned to i^{th} course registered in the semester under consideration

9.2 CGPA

The Cumulative Grade Point Average (CGPA) shall be computed after successful completion of the programme. The CGPA shall be expressed in different flavours to reflect B.Tech. of 161 credits, B.Tech. with Research Honours of 181 credits, and different Add-on provisions up to 20 credits leading to Minor / Add-on Diploma/ Add-on certification.

Accordingly, the computations will be as below:

$$\mathrm{CGPA} = \frac{\sum_{j=1}^{m} C_j P_j}{\sum_{j=1}^{m} C_j}$$

Where

m = total number of courses prescribed for the completion of the programme

P_j = grade points secured for the j^{th} course.

C_j = the number of credits assigned to j^{th} course

and $\sum C_j = 161$

$\sum C_j = 181$ for CGPA calculations in case of B.Tech. with Research honours

$\sum C_j = 20$ for CGPA calculations of specialization part in case of B.Tech. with honours

$\sum C_j = 181$ for CGPA calculations in case of B.Tech. with honours

$\sum C_j = 20$ for CGPA calculations of minor part in case of B.Tech. with Minor

$\sum C_j = 181$ for CGPA calculations in case of B.Tech. with Minor

$\sum C_j = 20$ for CGPA calculations of Add-on Diploma part in case of B.Tech. with Add-on Diploma

$\sum C_j = 181$ for CGPA calculations in case of B.Tech. with Add-on Diploma

$\sum C_j < 20$ for CGPA calculations of Add-on certification part in case of B.Tech. with Add-on certification

$\sum C_j < 181$ for CGPA calculations in case of B.Tech. with Add-on certification

Percentage equivalence of SGPA & CGPA = (SGPA or CGPA) X10

10. AWARD OF CLASS

The students who have become eligible for award of degree shall be classified based on their CGPA secured, as per the Table (10) given below:

Table 10 : Class/ Division information.

Sl. No.	CGPA	Class / Division
1	7.0 and above	First Class with Distinction
2	6.0 and above but less than 7.0	First Class
3	5.0 and above but less than 6.0	Second Class
4	4.0 and above but less than 5.0	Pass Class
5	Less than 4.0	Incomplete

a) For the purpose of rewarding the accomplishers with ranks and awards, toppers in each branch discipline are identified, based on their academic performance (CGPA) in the following categories:

i) Ranking in B.Tech.

ii) Ranking in B.Tech. with Research Honours

iii) Ranking in B.Tech. in Specialization

iv) Ranking in B.Tech. in Minor

b) In addition, the 'Chairman's gold medal' and other 'Endowment Awards' are awarded to the 'outstanding students' based on the overall performance which includes academic, co-curricular and extra-curricular activities, campus placements and competitive examinations. A committee appointed by the Vice-Chancellor will recommend the eligible student for the award, selected from the nominations received from the departments.

c) In addition, the institution may recognize exceptional performance such as music, dance, sports etc. and display of exceptional bravery from time to time.

d) Only such candidates who complete 161 credits (+20 credits) in the first 8 successive semesters shall be eligible to receive awards/ ranks.

e) The candidates availing spill over semesters will not be eligible for the award of merit scholarships.

11. AWARD OF DEGREE

On successful completion of prescribed requirements of the programme, the degree shall be conferred during the convocation of the VFSTR.

For the conferment of degree, the student has to fulfill the following requirements:

a) a bonafide student and undergone the course work of not less than four academic years and not more than seven academic years from the date of joining.

b) successfully completed all the courses as prescribed in the respective curriculum.

c) acquired a minimum eligible credits i.e. 161 credits for the award of B.Tech. degree.

d) obtained no due certificates as prescribed by VFSTR.

e) no in-disciplinary proceedings pending against him / her.

Consequent upon being convinced, following an enquiry, the Academic council may resolve to withdraw the degree / diploma / any other certification provided by the institute. The aggrieved may however prefer for a review of such decision by the Academic Council, citing cogent reasons for review or go in for an appeal to the, BoM of the institute.

12. LATERAL ENTRY AND LATERAL EXIT OPTIONS

12.1 Lateral Entry into II year of B.Tech.

The students who have completed their diploma programme in recognized institutes can be admitted into the third semester of the undergraduate programme. The committee constituted by Vice-Chancellor, will establish the eligibility of admission into different branches offered by VFSTR. The committee may recommend additional courses to be studied by the candidate, which students will complete during their first year of study at VFTSR however subject to a maximum of 25 credits per semester. To earn B.Tech. degree the student has to earn the same number of credits specified for regular B.Tech. in their 2nd, 3rd and 4th year of study. The student is eligible to register for Add-on credits as presented in section (1.5).

12.2 Inter- Institutional Credit Transfer

Students pursuing an undergraduate programme in other recognized Higher Education Institute (HEI) could be admitted into VFSTR without appearing for the entrance exam to continue their studies, subject to the approval by a committee constituted by Vice-Chancellor. In this case, the student shall furnish the transcripts or grade cards, syllabus copies, educational certificates, and other relevant documents while applying for admission. The committee constituted by Vice-Chancellor, will establish the equivalency based on the marks / credits obtained in the courses in the previous institute by following the Inter- Institutional Credit Transfer policy (Annexure-3).

VFSTR students also may earn credits by Inter- Institutional Credit Transfer.

12.3 Honorable exit with B.Sc. or Engineering Diploma

In line with NEP-2020, an optional exit is provided for a candidate who has earned a min. of 120 credits and has completed all the requirements up to the end of six semesters.

B.Sc. degree / Engineering Diploma will be awarded in respective branches of specialization. Candidates opted to exit with Engineering Diploma will also be awarded with B.Sc. equivalence certificate in case of disciplines existing in the University system.

In case the candidate fails to earn 120 credits, a suitable certification will be awarded during his / her exit from B.Tech. degree.

Semester-wise transcript and a consolidated transcript will be given to the candidates during their exit from the registered program.

Such a candidate who has exited can seek re-entry to complete B.Tech. by surrendering the B.Sc. Degree / Engineering Diploma. A committee constituted by Vice-Chancellor will scrutiny all such re-entry requests and recommend the plan of action. However, the max. duration of programme should be limited to seven years and further extension beyond the stipulated max. duration of study has to be approved by Academic Council, if the candidate appeals for an extension.

12.4 Volunteer ‘Drop’ with Sabbatical Semester option

A candidate may exercise his option to voluntarily exit from B.Tech. programme temporarily for a semester during the B.Tech. programme, by registering for a ‘DROP option’ in the beginning of the semester. The DROP can be exercised to take up special Internship / Innovation / Exploratory / Entrepreneurship / Advanced research / Start-up and such related activities. Under such circumstances a candidate can normally avail DROP over two successive semesters. Such ‘Drop’ semester will be identified as Sabbatical semesters.

Such a candidate has to pay the regular semester fee if such a Drop option is utilized during the first 8 semesters of B.Tech., and has to pay a nominal semester maintenance fee during the spillover period, if a candidate has not yet completed the credit requirements.

Upon returning from such a temporary exit, a candidate may continue his B.Tech. studies utilizing the provision of spillover period. A candidate may also submit a claim for Credit equivalence for the activities undertaken during the sabbatical period. The equivalence committee would evaluate and assess the academic equivalence of the work carried out and would recommend the credit equivalence and credit transfer to be granted together with the grades that could be attributed, if applicable. However, the max. duration of programme should be limited to seven years and further extension beyond the stipulated max. duration of study has to be approved by Academic Council, if the candidate appeals for an extension.

12.5 Volunteer 'Drop' with Semester Drop option

A candidate may exercise his option to voluntarily exit from B.Tech. programme temporarily for a semester during the B.Tech. programme, by registering for a 'DROP option' in the beginning of the semester to meet the family / personal exigencies. All the norms as mentioned in the section (12.4) shall be applicable for the candidates utilizing semester drop option.

13. CHANGE OF BRANCH DISCIPLINE

The students can request for change of branch after completing the first two semesters of study. One percent of the seats in each branch subject to a minimum of one seat is made available for accommodating such requests. These requests are considered subject to the following conditions:

a) Top one percent of the students in each branch based on CGPA at the end of the second semester subject to a minimum of one student in each branch are eligible for a change of branch.

b) If only one student is eligible from a branch and if he / she is not willing to change specialization, the opportunity can be availed by the second or third ranker of the branch in that order.

c) The seats must be available in the branch to which a student intends to change.

d) The student opting for a change should have completed all the credit requirements of the first two semesters.

e) The decision of the Institution is final in this regard.

All the above points are applicable for branch change cases based on the merit criterion. There may be branch change requests of another kind, from the students who are not able to cope up with the studies of the branch they are admitted into. All such requests are to be referred to a committee constituted for the purpose. The committee takes the decision based on the merit of each case, the availability of seats and various other factors. The decision of the committee in this regard is final. Because there is a difference of one or two courses in the first year curriculum among different specializations, when the student changes his specialization after completing first year, he / she is required to complete those first year courses of the new specialization which were not studied by him/her during the first year, which can be taken up in the immediate summer semester.

14. INTERPRETATION OF RULES

a) The academic rules and regulations should be read as a whole for the purpose of any interpretation.

b) For the matter(s) NOT covered herein above or for unforeseen circumstances, but arising during the course of the implementation of the above regulations. The Vice-Chancellor shall be authorized to remove the difficulties and decide upon the matters. The same shall be reported in the next meeting of Academic Council for ratification and subsequently informed to BoM.

c) The Institution may change or amend the academic rules and regulations or curriculum at any time, and the changes or amendments made shall be applicable to all the students with effect from the dates, notified by the Institution.

d) Procedure and explanation to any section can be floated by the office of Dean AAA as applicable from time to time with due approval by the chairman of Academic Council.

ANNEXURE -1

SUPPLEMENT REGULATION FOR PURSUING B.TECH. WITH RESEARCH HONOURS (SPECIALIZATION: XX)

A candidate willing to transform his / her B.Tech. programme into research orientation should meet the following terms and conditions.

1. He / she should register add-on credits for Honours specializing in a particular stream XX in his discipline of study.
2. He / she should have maintained a CGPA of 8.0 up to the end of 4th semester (II-year Even semester).
3. He / she should have successfully completed all the credit requirements up to the end of 4th semester.
4. He / she should indicate willingness to transfer his/her study from B.Tech. with Honours to B.Tech. with Research Honours during the Module-2 period of III year- Odd semester (5th Semester), when the applications are floated by the office of the Dean AAA for the purpose.
5. He / she has to appear for an aptitude test / interaction and the corresponding committee of experts constituted for the purpose has to recommend his / her name.
6. All recommended candidates may take up research project in a research lab / a research institution/ an institute of repute / a research organization or in the department of VFSTR.
7. All such candidates may compete for research internship support from the external research institutions or may be supported with research internship within VFSTR. This internship support will be for one full semester duration during his 8th semester extendable by one or two months during the Summer semester that falls in the sequel to 8th semester, in which case the extended period will be treated as 8th semester itself.
8. Such candidates are normally expected to pursue research in the area of specialization chosen for B.Tech. with Honours study.
9. Such candidates should register for a course on Research Methodology in III year- II semester as Open elective-3.
10. Such candidates should have identified a broad research problem by the end of 6th semester, as a part of his / her Module-2 requirement in the course on Research Methodology (Open elective-3).
11. In the 7th semester (IV year- I semester) he / she has to take up Research preliminaries for his / her Honours- 4 credit requirements. This will be a research project work by itself in the broad area of the proposed research problem. The candidate shall be able to complete a good literature survey, execute basic / preliminary implementations and by the end of module-2 of Honours – 4, he / she should have identified the specific research problem on which he / she will take up research during 8th semester. The entire course (Honours - 4) will be evaluated as per the requirements of a project work.
12. The candidate takes up the intensive research work during IV year – II semester for 16 credits (which is combined project work of 12 credits and Honours-5 of 4 credits). He / she should spend a minimum duration of 5 months on the research work. If required, he / she will be considered for research internship period in the extended period also. In fact, he/ she may conserve time by starting the research work immediately after the completion of 7th semester.

Thus a candidate should have successfully earned 161 + 20 = 181 credits following the aforementioned procedure to qualify for B.Tech. in YY Engineering with Research Honours (Specialization: XX).

ANNEXURE – 2(a)

SUPPLEMENT REGULATION FOR PURSUING DUAL B.TECH. + M.TECH. DEGREE

The proposal to institute Dual B.Tech. + M.Tech., in line with the practices in Institutions of National Importance, is to elevate the Gross Qualification Index (GQI) of India, and specifically to enhance the technological/ engineering competency of the Technocrats.

1. In order to attract the committed learners towards earning M.Tech. immediately after B.Tech. the following scheme is proposed:
2. Every candidate should express his / her provisional intent at the time of B.Tech. admission for pursuing Dual B.Tech. + M.Tech. degree. However, he / she can exercise his option to change his intent and final confirmation should be submitted by the end of III-year B.Tech. programme (VI semester) for Dual degree.
3. A candidate needs to be well informed about the advantages that he / she will reap by completing a Master's degree in one stretch as an immediate follow up to B.Tech. degree program. Such M.Tech. program could enhance his orientation to research studies and could elevate as a researcher.
4. To make the proposal very attractive, the scheme incorporated to effectively reduce the study period from 4+2 = 6 years to around five years only by exploiting the continuity of the study sustaining the tempo and through making use of two Summer semesters between III & IV year and between IV & V year, and still technically maintaining the requirement to complete 8 semesters of B.Tech. + 4 semesters of M.Tech. = 12 Semesters intact.

A candidate willing to continue Dual B.Tech. + M.Tech. instead of regular B.Tech. programme should follow the following terms, conditions and procedures:

1. He / she should be a candidate maintaining his studentship through proper registration process of his B.Tech. programme.
2. He / she can exercise his / her option to earn Add-on credits up to 20 along with his / her regular B.Tech. to receive B.Tech. with Honours / with Minor / with Add-on Diploma.
3. He / she should have completed all the credit requirements up to the end of 5th semester successfully with a CGPA of at least 7.5. the candidate is expected to likely to complete 6th semester successfully and to maintain the CGPA of at least 7.
4. He / she should indicate his willingness to continue his study in Dual B.Tech. + M.Tech. during the Module-2 period of III-year – Even semester (6th semester). When the call is given by the office of the Dean AAA for the purpose.
5. He / she has to appear for an aptitude test / interaction and the corresponding committee of experts constituted for the purpose has to recommend his / her name.
6. The candidate formally enrolls for Dual B.Tech. + M.Tech. by the end of VI semester before the semester-end summative assessment period of VI semester.
7. The study pattern will be as follows:

Period	M.Tech. semester count	Cumulative semester count	Expected study coverage
Sandwiched summer semester between III & IV year	I semester	7th semester	14 credits of M.Tech. aligning with M.Tech. Odd semester
Regular B.Tech. VII semester		8th semester	As per B.Tech VII semester curriculum

Regular VIII semester	II semester	● Internship / Project 12 credits of B.Tech. is suspended. ● 4 credits of Honours / Minor would continue in 9th semester	16 credits of M.Tech. aligned with M.Tech. even semester
Sandwiched summer semester between IV & V year	III semester	10th semester	Remaining core / electives required in M.Tech. - 14 credits
Regular odd semester in V year	Suspended B.Tech. VIII semester credits as a preamble to M.Tech. project work	11th semester	● Internship / project work 12 credits corresponding to B. Tech. ● 4 credits of IV semester of M. Tech. on Research methodology ● 4 credits of IV semester of M. Tech. on preliminary Research work with a project
Regular even semester in V year	IV semester	● 12th semester ● If required IV semester may be extended by 1 or 2 months to complete the research work.	Research project of M. Tech. 16 credits.

Thus the credit requirements specified for both B.Tech. and M.Tech. are satisfied along with the 12 semester requirement for B.Tech. and M.Tech. put together

- B.Tech. degree = 161 + 20 Add-on credits (opted)
- M. Tech degree = 68 (14+ 16 +14 +08 +16 credits)

A higher research orientation is created.

8. In the convocation after V year the candidate receives both the degrees B. Tech. & M. Tech. under dual mode.

9. The candidate should not opt for campus placements during III year / IV year beginning. He / she will be permitted to pursue for campus placement during 4th year end / 5th year.

10. Such a candidate may try for fellowship support during V year if he / she gets qualified in GATE; such candidates may also be eligible for support consideration by VFSTR during V year.

11. Such a candidate during his / her V year could be considered as Teaching cum Research Assistant and suitable compensation may be provided for his / her association as a Teaching Assistant in the VFSTR campus.

The proposed Dual B.Tech. + M.Tech. may be offered as a customized Dual B.Tech. + M.Tech. program jointly with an Industry for the candidates freshly placed in that industry. Then the orientation would be towards on-Job training, Advanced Industry skills and Job basis internship & project work.

ANNEXURE – 2(b)

SUPPLEMENT REGULATION FOR PURSUING DUAL B.Tech. + MBA DEGREE

The proposal to institute Dual B.Tech. + MBA, in line with the practices in Institutions of National Importance, is to elevate the Gross Qualification Index (GQI) of India, and specifically to enhance the technological / management competency of the Technocrats. In order to attract the committed learners towards earning MBA immediately after B.Tech. the following scheme is proposed:

1. Every candidate should express his / her provisional intent at the time of B.Tech. admission for pursuing Dual B.Tech. + MBA degree. However, he / she can exercise his option to change his intent and final confirmation should be submitted during the III semester of B.Tech. programme for Dual degree.
2. A candidate needs to be well informed about the advantages that he / she will reap by completing a Master's degree in one stretch as an immediate follow up to B.Tech. degree program. Such MBA program could enhance his orientation to management studies and could elevate as a Manager / Entrepreneur.
3. To make the proposal very attractive, the scheme incorporated to effectively reduce the study period from 4+2 = 6 years to around five years only by exploiting the continuity of the study sustaining the tempo and through making use of two Summer semesters between III & IV year and between IV & V year, and still technically maintaining the requirement to complete 8 semesters of B.Tech. + 4 semesters of MBA = 12 Semesters intact.

A candidate willing to continue Dual B.Tech. + MBA instead of regular B.Tech. programme should follow the following terms, conditions and procedures:

1. He / she should be a candidate maintaining his studentship through proper registration process of his B.Tech. programme.
2. He / she should opt Engineering Management stream as Minor during regular B.Tech and the 20 credits earned in Minor will be considered for extension into dual B.Tech. + MBA.
3. He / she should have completed all the credit requirements up to the end of 3rd semester successfully and the candidate is expected to likely to complete 4th semester successfully.
4. He / she should indicate his willingness to continue his study in Dual B.Tech. + MBA during the Module-2 period of II-year – odd semester (3th semester). When the call is given by the office of the Dean AAA for the purpose.
5. The study pattern will be as follows:

Period	MBA semester count	Cumulative semester count	Expected study coverage
Regular B.Tech. IV Semester	-	4th semester	4 credits of Management Minor
Regular B.Tech. V Semester	-	5th semester	4 credits of Management Minor
Regular B.Tech. VI Semester	-	6th semester	4 credits of Management Minor
Sandwiched summer semester between III & IV year	I Year I semester	7th semester	16 credits of MBA
Regular B.Tech. VII semester		8th semester	4 credits of Management Minor
Regular B.Tech. VIII semester		9th semester	4 credits of Management Minor
Sandwiched summer semester between IV & V year	I Year II Semester	10th semester	16 credits of MBA

Regular odd semester in V year	II Year I semester	11th semester	20 credits of MBA
Regular even semester in V year	II Year II semseter	12th semester	Internship of MBA 12 credits.

Thus the credit required of both B.Tech. and MBA are satisfied along with the 12 semester requirement for B.Tech. and MBA put together

- B.Tech. degree = 161
- MBA degree = 64 (16+ 16 +20 +12 credits) +20 Add-on credits (opted in B.Tech. for Management minor)

  Total credits = 161 + 20 + 64 = 245

6. In the convocation after V year the candidate receives both the degrees B. Tech. & MBA under dual mode.
7. The candidate should not opt for campus placements during III year / IV year beginning. He / she will be permitted to pursue for campus placement during 4th year end / 5th year.

Annexure – 3

INTER – INSTITUTE CREDIT TRANSFER

Students pursuing an undergraduate programme in other recognized Higher Education Institute (HEI) could be admitted into VFSTR without appearing for the entrance exam to continue their studies, subject to the approval by a committee constituted by Vice-Chancellor. In this case, the student shall furnish the transcripts or grade cards, syllabus copies, educational certificates, and other relevant documents while applying for admission. The committee constituted by Vice-Chancellor, will establish the equivalency based on the marks/credits obtained in the courses in the previous institute by following the admission policy as mentioned below.

PROCEDURE

- Admissions are only at the beginning of an academic year and depends on availability of vacancies in the respective program.
- The courses studied by the candidates elsewhere will be mapped with the courses offered by VFTSR and the committee may suggest for study of additional courses for maintaining credit equivalency or for satisfying pre-requisites. The committee may also recommend the plan of study for the courses identified.
- After admission, the student has to submit the following documents
  - Transfer Certificate
  - Grade card / Marks Memo
  - “No objection certificate (NOC)” from the concerned Institute / University where the candidate completed part of the programme.
- Provisional admission of the candidates will be confirmed only after verification of the authenticity of the certificates from the concerned authorities.
- Min. 50% of the credits have to be earned at VFSTR.
- Courses which are completed by the candidate in the previous institute will be transferred as recommended by the equivalence committee under credit cum grade transfer. Inclusive these credits the candidate has to complete the credit requirements as per VFSTR.
- Consolidated grade card / transcript will clearly show the credits under credit transfer and corresponding CGPA and credits earned in VFSTR and corresponding CGPA. Over all CGPA may also be indicated.

EXTENSION:

VFSTR candidates may also be permitted to earn credits up to max of 20% by the way of credit transfer from taking up the courses from other institutes of repute.

Annexure – 4
COURSE STRUCTURE - R22

I Year I Semester

S.No.	Course Title	L	T	P	C	Course category
1	Mathematics – I	3	0	2	4	Basic Sciences
2	Physics / Chemistry / Department Subject	2	0	2	3	Basic Sciences
3	Basic of Electrical & Electronics Engineering/ Engineering Graphics	2	0	2	3	Basic Engineering
4	Workshop & Basic Engineering Products - Program specific	0	0	6	3	Basic Engineering
5	Programming in C	2	0	4	4	Basic Engineering
6	English Proficiency & Communication Skills (PET)	0	0	2	1	Humanities
7	Physical Fitness, Sports & Games – I	0	0	3	1	Binary grade
8	Constitution of India	0	2	0	1	Humanities
Total		**9**	**2**	**21**	**20**	
		32 Hrs				

I Year II Semester

S.No.	Course Title	L	T	P	C	Course category
1	Mathematics – II	3	0	2	4	Basic Sciences
2	Physics / Chemistry / Department Subject #	2	0	2	3	Basic Sciences
3	Basic of Electrical & Electronics Engineering/Engineering Graphics	2	0	2	3	Basic Engineering
4	Competitive coding (Basic)	0	1	3	2	Basic Engineering
5	Technical English Communication	2	0	2	3	Humanities
6	Professional core – 1	3	1	0	4	Professional core
7	Physical Fitness, Sports & Games – II	0	0	3	1	Binary grade
8	Orientation Session	1	4	0	3	Humanities
Total		**13**	**6**	**14**	**23**	
		33 Hrs				

# Department Subject is extension of Basic sciences

II Year I Semester

S.No.	Course Title	L	T	P	C	Course category
1	Maths related course (P&S, PTSP……….)	3	0	2	4	Basic Sciences
2	Environmental Studies	1	0	0	1	Basic Sciences
3	Data Structures	2	0	4	4	Basic Engineering
4	Management Science / Principles of Management	2	0	2	3	Humanities
5	Professional core – 2	3	0	2	4	Professional core
6	Professional core – 3	3	0	2	4	Professional core
7	Professional core – 4	2	0	2	3	Professional core
8	Life Skills	0	0	2	1	Binary grade
Total		**16**	**0**	**16**	**24**	
9	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication with good impact factor (Only 2 students can claim 1 paper /patent). These credits maybe earned on or before the end of IV semester	0	0	0	**1**	Floating credits Binary grade
Total		**16**	**0**	**16**	**25**	
		32 Hrs				

II Year II Semester

S.No.	Course Title	L	T	P	C	Course category
1	Competitive coding (Advanced)	0	0	2	1	Basic Engineering
2	Professional Communication (BEC Certification)	0	0	2	1	Humanities
3	Professional core – 5	3	0	2	4	Professional core
4	Professional core – 6	3	0	2	4	Professional core
5	Professional core – 7	2	0	2	3	Professional core
6	Department Elective – 1	2	0	2	3	Department Elective
7	Open Elective – 1	2	0	2	3	Open Elective
8	Life Skills	0	0	2	1	Binary grade
Total		**12**	**0**	**16**	**20**	
9	**Minor / Honors – 1**	3	0	2	**4**	
Total		**15**	**0**	**18**	**24**	
		33 Hrs				

III Year I Semester

S.No.	Course Title	L	T	P	C	Course category
1	Soft Skills Laboratory	0	0	2	1	Humanities
2	Professional core – 8	3	0	2	4	Professional core
3	Professional core – 9	3	0	2	4	Professional core
4	Professional core – 10	3	0	2	4	Professional core
5	Department Elective – 2	2	0	2	3	Department Elective
6	Open Elective – 2	2	0	2	3	Open Elective
7	Industry interface course (Modular course)	1	0	0	1	Elective
8	Inter-departmental Project / Course	0	0	2	0	Project
Total		**14**	**0**	**14**	**20**	
9	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication with good impact factor (Only 2 students can claim 1 paper /patent). These credits maybe earned on or before the end of VI semester	0	0	0	1	Floating credits Binary grade
10	**Minor / Honors – 2**	3	0	2	**4**	
Total		**17**	**0**	**16**	**25**	
		33 Hrs				

III Year II Semester

S.No.	Course Title	L	T	P	C	Course category
1	Quantitative aptitude & Logical reasoning	1	1	0-	2	Humanities
2	Professional core – 11	3	0	2	4	Professional core
3	Professional core – 12	2	0	2	3	Professional core
4	Department Elective – 3	2	0	2	3	Department Elective
5	Department Elective – 4	2	0	2	3	Department Elective
6	Open Elective – 3	2	0	2	3	Open Elective
7	Inter-departmental Project/Course	0	0	2	2	Project
	Total	**12**	**1**	**12**	**20**	
8	**Minor / Honors – 3**	3	0	2	**4**	
Total		**15**	**1**	**14**	**24**	
		30 Hrs.				

IV Year I Semester

S.No.	Course Title	L	T	P	C	Course category
1	Professional core – 13	3	0	2	4	Professional core
2	Professional core – 14	3	0	2	4	Professional core
3	Department Elective – 5	2	0	2	3	Department Elective
4	Department Elective – 6	2	0	2	3	Department Elective
5	Department Elective – 7	2	0	2	3	Department Elective
6	Department Elective – 8	2	0	2	3	Department Elective
Total		**14**	**0**	**12**	**20**	
7	**Minor / Honors – 4**	3	0	2	4	
Total		**17**	**0**	**14**	**24**	
		31 Hrs				

IV Year II Semester

S.No.	Course Title	L	T	P	C	Course category
1	Internship / Project Work	0	2	22	12	Project
Total		**24 Hrs**			**12**	
2	**Minor / Honors – 5 (for project)**	**0**	**2**	**6**	**4**	Theory course may be also offered
Total		**32 Hrs**			**16**	

# for interaction between guide and students.

B.Tech. INFORMATION TECHNOLOGY

TABLE OF CONTENTS

FOREWORD

Information Technology (IT) is transforming our world, in a myriad way. Recent developments in IT technologies such as Artificial Intelligence, Machine Learning, Artificial Neural Networks, Deep Learning, Data Science, Computer Vision, Bigdata Analytics, Internet of Things, Cyber security, Cloud computing and Black Chain Technologies are becoming part of many engineering applications used in day to day life. It is changing the way we do business, the way we learn, the way we communicate, and even the way we entertain ourselves. The discipline of Information Technology addresses the integration, design, deployment, management of computing, networking resources and services, as well as the development of technology infrastructures in organizations. The Indian IT industry has undergone a rapid evolution, as it has kept abreast of what the global markets require in terms of products and services. The country has now positioned itself as an IT hub, a sourcing destination for IT products and services that spell trust, high quality, and cost-effectiveness. The IT industry has shaped itself into a process-oriented, best practices-focused and skill-rich entity that has found favor with global customers. IT specialists solve complex software and hardware problems that require fundamental knowledge, competencies and utilize the processes of needs assessment, technology transfer, and user support.

R22 curriculum comprises of:

- *Six months Industry Internship to make students industry ready.*
- *Projects of inter-disciplinary.*
- *Revision in tune with National Education Policy 2020.*
- *Multiple entry and multiple exit options.*
- *Regular degree along with honours / minor degree.*
- *Module wise course syllabus.*
- *More emphasis on problem solving skills.*
- *One industry interface course (modular course) with industry support.*
- *Enhanced skill based courses for improving employment opportunities.*
- *Advanced courses like Artificial Intelligence, Machine learning, Artificial Neural Networks, Deep Learning, Cloud Computing, Big Data Analytics and Internet of Things.*
- *Practice and tutorial sessions embedded into as many courses as possible.*

The Board of Studies consisting eminent personalities along with experienced faculty members of the university have designed the curriculum to offer knowledge and skill of information technology on the above mentioned areas. The curriculum includes concepts with skill based tasks through integrated laboratory and activities combined with theory. The department aims to make graduates ready for the industrial needs

External BoS Members:

1. Dr.C.Raghavendra Rao, Professor, University of Hyderabad, Hyderabad.
2. Dr. R.B.V. Subramaanyam, Professor & HOD, Department of CSE, NIT Warangal.
3. Dr. B. Ramesh Babu, Sr. Asst. Professor, Department of CSE, MNIT Jaipur.

I thank all the BOS Members, Academic Council Members, and University authorities especially Dean AAA and honourable Vice Chancellor for encouraging and supporting us in designing this innovative curriculum for IT students.

Dr. N.Veeranjaneyulu
HOD, IT

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be UNIVERSITY)
-Estd. u/s 3 of UGC Act 1956

VISION

To evolve into a Centre of Excellence in Science & Technology through creative and innovative practices in teaching – learning, towards promoting academic achievement and research excellence to produce internationally accepted, competitive and world class professionals who are psychologically strong & emotionally balanced, imbued with social consciousness & ethical values.

MISSION

To provide high quality academic programmes, training activities, research facilities and opportunities supported by continuous industry - institute interaction aimed at promoting employability, entrepreneurship, leadership and research aptitude among students and contribute to the economic and technological development of the region, state and nation.

Department of INFORMATION TECHNOLOGY

VISION OF THE DEPARTMENT

To emerge as a leading hub of excellence in Information Technology by fostering innovative teaching-learning practices, advancing cutting-edge research, and nurturing globally competitive IT professionals with strong ethical values, social responsibility.

MISSION OF THE DEPARTMENT

- To deliver high-quality education in Information Technology through robust academic programs that blend theory and practical application.
- To promote interdisciplinary research, innovation, and entrepreneurship through collaborations with industry and research organizations.
- To instil leadership skills, ethical values, social awareness, and emotional resilience and prepare graduates as responsible IT professionals.

B. Tech. INFORMATION TECHNOLOGY

Programme Educational Objectives (PEOs)

PEO1: To excel in their career as competent software engineer in IT and allied organizations, enabling them to design and develop effective and efficient IT solutions in real-world applications.

PEO2: To demonstrate research temper and address complex problems in rapidly evolving IT industry, preparing them as entrepreneurs.

PEO3: To integrate ethical values, social responsibility, and sustainability in their IT practices and exhibit leadership skills.

Programme Specific Outcomes (PSOs)

PSO1: Provide solutions in the area of database management, software design and computing systems using machine intelligence.

PSO2: Design, implement and manage secure computer networks and information systems by applying cybersecurity.

PSO3: Apply software engineering principles, agile methodologies and project management techniques to develop and deliver high quality IT products.

Programme Outcomes (POs)

PO1: **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamen tals, and an engineering specialization to the solution of complex engineering problems.

PO2: **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering prob lems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: **Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

COURSE STRUCTURE - R22

I Year I Semester

Course Code	Course Title	L	T	P	C
22MT103	Linear Algebra and Ordinary Differential Equations	3	2	-	4
22PY105	Semiconductor Physics and Electromagnetics	2	-	2	3
22EE101	Basics of Electrical and Electronics Engineering	2	-	2	3
22IT102	IT Workshop and Tools	-	2	4	3
22TP103	Programming in C	2	-	4	4
22EN102	English Proficiency and Communication Skills	-	-	2	1
22SA101	Physical Fitness, Sports and Games – I	-	-	3	1
22TP101	Constitution of India	-	2	-	1
Total		**9**	**6**	**17**	**20**
		32 Hrs			

I Year II Semester

Course Code	Course Title	L	T	P	C
22MT106	Algebra	3	2	-	4
22MT107	Discrete Mathematical Structures	2	2	-	3
22TP104	Basic Coding Competency	-	1	3	2
22ME101	Engineering Graphics	2	-	2	3
22EN104	Technical Communication English	1	2	2	3
22IT101	Web Technologies	1	2	2	3
22SA103	Physical Fitness, Sports and Games – II	-	-	3	1
	Total	**9**	**9**	**12**	**19**
		30 Hrs			

I Year I / II Semester

Course Code	Course Title	L	T	P	C
22SA102	Orientation Session	-	-	6	3

COURSE STRUCTURE - R22

II Year I Semester

Course Code	Course Title	L	T	P	C
22ST202	Probability and Statistics	3	2	-	4
22CT201	Environmental Studies	1	1	-	1
22TP201	Data Structures	2	2	2	4
22MS201	Management Science	2	2	-	3
22IT201	Database Systems	3	-	2	4
22IT202	Digital Logic Design	2	2	-	3
22IT203	Object Oriented Programming	3	-	2	4
22SA201	Life Skills - I	-	-	2	1
	Total	16	9	8	24
	NCC / NSS / SAC / E-cell / Student Mentoring/ Social activities/ Publication	-	-	-	1
	Total	**16**	**9**	**8**	**25**
		33 Hrs			

II Year II Semester

Course Code	Course Title	L	T	P	C
22TP203	Advanced Coding Competency	-	-	2	1
22TP204	Professional Communication	-	-	2	1
22IT204	Design and Analysis of Algorithms	3	-	2	4
22IT205	Operating Systems	3	-	2	4
22IT206	Python Programming	2	-	2	3
	Department Elective-I	2	2	-	3
	Open Elective – 1	2	2	-	3
22SA202	Life Skills - II	-	-	2	1
	Total	**12**	**4**	**12**	**20**
	Minor / Honours - 1	3/0	0/4	2/4	4
		15	**4**	**14**	**24**
	Total	**33/36 Hrs**			

COURSE STRUCTURE - R22

III Year I Semester

Course Code	Course Title	L	T	P	C
22TP301	Soft Skills Lab	-	-	2	1
22IT301	Computer Networks	3	-	2	4
22IT302	Data Mining Techniques	3	-	2	4
22IT303	Software Engineering	3	-	2	4
	Department Elective – 2	2	2	-	3
	Open Elective – 2	2	2	-	3
22IT304	Inter-Disciplinary Project – Phase I	-	-	2	-
22IT305	Industry Interface Course (Modular course)	1	-	-	1
	Total	**14**	**4**	**10**	**20**
	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication	-	-	-	1
	Minor / Honors – 2	3/0	0/4	2/4	4
	Total	**17**	**4**	**12**	**25**
		33/36 Hrs			

III Year II Semester

Course Code	Course Title	L	T	P	C
22TP302	Quantitative Aptitude and Logical Reasoning	1	2	-	2
22IT306	Cloud Computing	3	-	2	4
22IT307	Machine Learning	3	-	2	4
	Department Elective – 3	2	2	-	3
	Department Elective – 4	2	2	-	3
	Open Elective – 3	2	2	-	3
22IT308	Inter-Disciplinary Project – Phase II	-	-	2	2
	Total	**13**	**8**	**6**	**21**
	Minor / Honors – 3	3/0	0/4	2/4	4
	Total	**16**	**8**	**8**	**25**
		32/35 Hrs			

COURSE STRUCTURE - R22

IV Year I Semester

Course Code	Course Title	L	T	P	C
22IT401	Cryptography and Network Security	3	-	2	4
22IT402	Internet of Things	3	-	2	4
	Department Elective – 5	3	-	2	4
	Department Elective – 6	3	-	2	4
	Department Elective – 7	3	2	-	4
	Total	**15**	**2**	**8**	**20**
	Minor / Honors – 4	3/0	0/4	2/4	4
	Total	**18**	**2**	**10**	**24**
		30/33 Hrs			

IV Year II Semester

Course Code	Course Title	L	T	P	C
22IT403 / 22IT404	Project Work / Internship	-	2#	22	12
	Total	**24**			**12**
	Minor / Honours – 5 (for project)	-	2/4	6/4	**4**
	Total	**32 Hrs**			**16**

# for interaction between Guide and students

L=Lecture; T= Tutorial; P= Practical; C=Credits

COURSE STRUCTURE - R22

Department Electives

C Code	Course Title	L	T	P	C
22IT827	Open Source Web Technologies	3	-	2	4
22IT802	Advanced Web Technologies	3	-	2	4
22IT823	Mobile Computing	2	2	-	3
22IT822	Mobile Application Development	2	2	-	3
22IT829	Semantic Web and Social Networks	2	2	-	3
22IT818	Full Stack Technologies	3	-	2	4
22IT808	Computer Graphics	2	2	-	3
22IT815	Digital Image Processing	2	2	-	3
22IT828	Pattern Recognition	2	2	-	3
22IT819	Image and Video Analytics	3	-	2	4
22IT810	Computer Vision	3	-	2	4
22IT820	Medical Image Analysis	3	-	2	4
22IT803	Artificial Intelligence	2	2	-	3
22IT805	Big Data Analytics	3	-	2	4
22IT813	Data Science using Python	3	-	2	4
22IT804	Artificial Neural Networks	2	2	-	3
22IT814	Deep Learning	3	-	2	4
22IT824	Natural Language Processing	3	-	2	4
22IT812	Data Communications	3	-	2	4
22IT826	Network Protocols	2	2	-	3
22IT831	Wireless Networks	2	2	-	3
22IT811	Cyber Security	2	2	-	3
22IT825	Network Programming	3	-	2	4
22IT833	Block Chain Technologies	2	2	-	3
22IT809	Computer Organization and Architecture	2	2	-	3
22IT817	Formal Languages and Automata Theory	3	2	-	4
22IT807	Compiler Design	3	2	-	4
22IT801	Advanced Computer Architecture	2	2	-	3
22IT816	Distributed Systems	2	2	-	3
22IT821	Microprocessors and Microcontrollers	3	2	-	4

COURSE STRUCTURE - R22

Honours

C Code	Course Title	L	T	P	C
22IT951	Artificial Neural Networks	3	-	2	4
22IT953	Deep Learning	3	-	2	4
22IT952	Computer vision	3	2	-	4
22IT954	image and video Analytics	3	-	2	4
22IT955	Natural Language Processing	3	-	2	4
22IT956	Coding Competency	-	4	4	4
22IT957	Advanced Web Technologies	-	4	4	4
22IT963	DevOps Technologies	-	4	4	4
22IT967	Cloud Full Stack Development	-	4	4	4
22IT960	Capstone Project	-	4	4	4

I YEAR

B.Tech.

INFORMATION TECHNOLOGY

I SEMESTER

22MT103	-	Linear Algebra and Ordinary Differential Equations
22PY105	-	Semiconductor Physics and Electromagnetics
22EE101	-	Basics of Electrical and Electronics Engineering
22IT102	-	IT Workshop and Tools
22TP103	-	Programming in C
22EN102	-	English Proficiency and Communication Skills
22SA101	-	Physical Fitness, Sports and Games – I
22TP101	-	Constitution of India

II SEMESTER

22MT106	-	Algebra
22MT107	-	Discrete Mathematical Structures
22TP104	-	Basic Coding Competency
22ME101	-	Engineering Graphics
22EN104	-	Technical Communication English
22IT101	-	Web Technologies
22SA103	-	Physical Fitness, Sports and Games – II
22SA102	-	Orientation Session

COURSE CONTENTS

I SEM & II SEM

22MT103 LINEAR ALGEBRA AND ORDINARY DIFFERENTIAL EQUATIONS

Source: https://www.amazon.com/Differential-Equations/dp/B01H30X2JA

Hours Per Week :

L	T	P	C
3	2	-	4

PREREQUISITE KNOWLEDGE: Basics of matrices, Differentiation and Integration.

COURSE DESCRIPTION AND OBJECTIVES:

The goal of this course is to build a grasp of the principles of mathematics through matrices, differential equations and applications that serves as an essential tool in several engineering applications.

MODULE-1

UNIT-1 **12L+8T+0P=20 Hours**

MATRICES:

Definition of matrix; Types of matrices; Algebra of matrices, adjoint of a matrix, inverse of a matrix through adjoint and elementary row operations, Rank of a matrix, Echelon form, Normal form.

Eigen values and Eigen vectors (up to 3 x 3 matrices only) and properties (without proofs).

UNIT-2 **12L+8T+0P=20 Hours**

APPLICATIONS OF MATRICES:

Consistency of system of linear equations, Solution of system of linear equations having unique solution and involving not more than three variables by Gauss elimination method and Gauss Jordan method.

Cayley-Hamilton theorem (without proof), Power of a matrix, Inverse of a matrix.

Strength of materials and strength of beams using Eigen value and Eigen vectors.

PRACTICES:

- Compute inverse of a matrix if exists.
- Explain with suitable examples how rank of matrix is independent of the elementary operations.
- Explain with suitable examples how rank of matrix is unique.
- Discuss with suitable examples when eigen values and eigen vectors are possible for a matrix.
- Discuss the possibility of solution of a system of equations.
- Discuss when inverse and power of a matrix exist using Cayley-Hamilton theorem.

MODULE-2

UNIT-1 **12L+8T+0P=20 Hours**

ORDINARY DIFFERENTIAL EQUATIONS (ODE):

First Order Differential Equations: Introduction to ODE, variable separable method, homogenous and non-homogenous differential equations, linear differential equations, Bernoulli’s equations.

Second Order Differential Equations: Linear differential equations with constant coefficients with RHS of the form eax, xn, sin(ax) or cos(ax).

SKILLS:

- ✓ *Find rank of a matrix using different methods.*
- ✓ *Compute the eigen values and eigen vectors of a matrix.*
- ✓ *Find analytical solution of a differential equation using appropriate method.*
- ✓ *Demonstrate any one numerical method to solve differential equation.*

UNIT-2 **12L+8T+0P=20 Hours**

APPLICATIONS OF ODE:

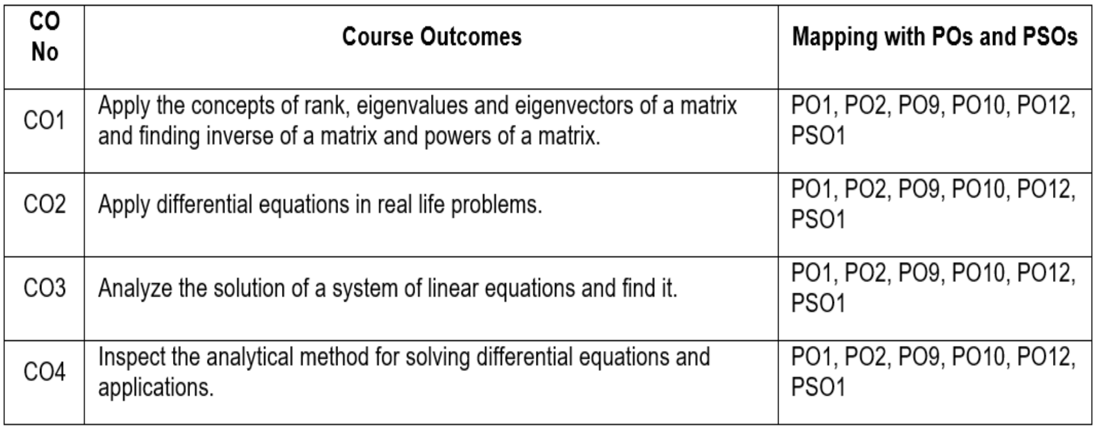
Applications of ODE: Newton's law of cooling, Law of natural growth and decay, LR Circuit.

PRACTICES:

- Check the order and degree of an ODE.
- Find solution for any four ordinary differential equations by applying suitable method.
- Find numerical solution for any four ordinary differential equations by applying suitable method.
- Discuss some applications of ODE.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the concepts of rank, eigenvalues and eigenvectors of a matrix and finding inverse of a matrix and powers of a matrix.	PO1, PO2, PO9, PO10, PO12, PSO1
CO2	Apply differential equations in real life problems.	PO1, PO2, PO9, PO10, PO12, PSO1
CO3	Analyze the solution of a system of linear equations and find it.	PO1, PO2, PO9, PO10, PO12, PSO1
CO4	Inspect the analytical method for solving differential equations and applications.	PO1, PO2, PO9, PO10, PO12, PSO1

TEXTBOOK:

1. 1. N. P. Bali, K. L. Sai Prasad, "A Textbook of Engineering Mathematics I, II, III", 2nd Edition Universal Science Press, New Delhi, 2018.
2. B. S. Grewal, "Higher Engineering Mathematics", 44 Edition, Khanna Publishers, 2018.

REFERENCE BOOKS:

1. Erwin Kreyszig, "Advanced Engineering Mathematics", 10th Edition, John Wiley & Sons, Inc, 2010.
2. H. K. Dass and Er. RajanishVerma, "Higher Engineering Mathematics", 3rd revised edition, S. Chand & Co., 2015.
3. B. V. Ramana, "Advanced Engineering Mathematics", TMH Publishers.
4. T. K.V. Iyengar et al, "Engineering Mathematics, I, II, III", S. Chand & Co., New Delhi,2018.

22PY105 SEMICONDUCTOR PHYSICS AND ELECTROMAGNETICS

Hours Per Week :

L	T	P	C
2	-	2	3

PREREQUISITE KNOWLEDGE: Basics of vectors and semiconductors.

COURSE DESCRIPTION AND OBJECTIVES:

This course ensures commensurable understanding of electrostatics and magneto statics. It enunciates the electron dynamics in solids through the conceptual grasp of principles of quantum mechanics. This embark perspective outlook on optoelectronic devices and optical fibres in the backdrop of semiconductor physics.

MODULE-1

UNIT-1 **10L+0T+10P = 20 Hours**

ELECTROSTASTICS AND MAGNETOSTASTICS:

Electrostatics: Introduction to Vector analysis, Computation of electric field and potential due to Point charge, linear charge density, surface charge density, bulk charge density, Coulomb's law, Electric field due to line of charges, Gauss law, Differential Form of Gauss law, Applications, Electric field due to a charged sphere – inside, on the surface, and outside, Electric field due to a spherical shell- inside and outside.

Magnetostatics: Introduction to magnetic force – Lorentz force, Biot-Savart's law, Magnetic field due to a linear conductor – magnetic field due to a circular loop –Ampere's law, Faraday's law in integral form; Lenz's law, Maxwell's equations – correction to Ampere's law.

UNIT-2 **6L+0T+6P = 12 Hours**

QUANTUM MECAHNICS AND FREE ELECTRON THEROY:

Quantum mechanics: Introduction to Quantum mechanics; Concepts of wave and particle duality of radiation; de Broglie's concepts of matter waves, Schrödinger's time-independent wave equation – Eigen values and Eigen functions; Particle confined in a one-dimensional infinite Potential square well.

Free electron theory of solids: Classical and Quantum free electron theory of metals; Fermi- Dirac distribution; Density of states – derivation -Bloch's Theorem (Qualitative); Classification of solids based on energy bands.

PRACTICES:

- Photoelectric effect-Determination of plancks constant.
- Stewart & Gee's Experiment- Study of magnetic field along the axis of a current carrying coil.
- Melde's Experiment - determination of the frequency of tuning fork.
- Hall Effect - Determination of Hall coefficient.

MODULE-2

UNIT-1 **8L+0T+8P = 16 Hours**

SEICONDUCTOR PHYSICS AND OPTOELECTRONICS:

Introduction, Classification of Semiconductors, Direct and indirect band gap semiconductors, Intrinsic semiconductors; Variation of Intrinsic carrier concentration with temperature, Fermi level, and conductivity; Extrinsic semiconductor, the effect of temperature on carrier concentration in extrinsic semiconductors, Band diagrams of extrinsic semiconductors; Hall effect, Classification of optoelectronic devices; Photo voltaic cell, LED.

SKILLS:

- ✓ *Able to compute the electric and magnetic field and potentials in different applications*
- ✓ *Apply the quantum laws to understand the electron dynamics of solids*
- ✓ *Realizing the importance of optoelectronic devices*

UNIT-2 **8L+0T+8P = 16 Hours**

LASERS AND OPTICAL FIBERS:

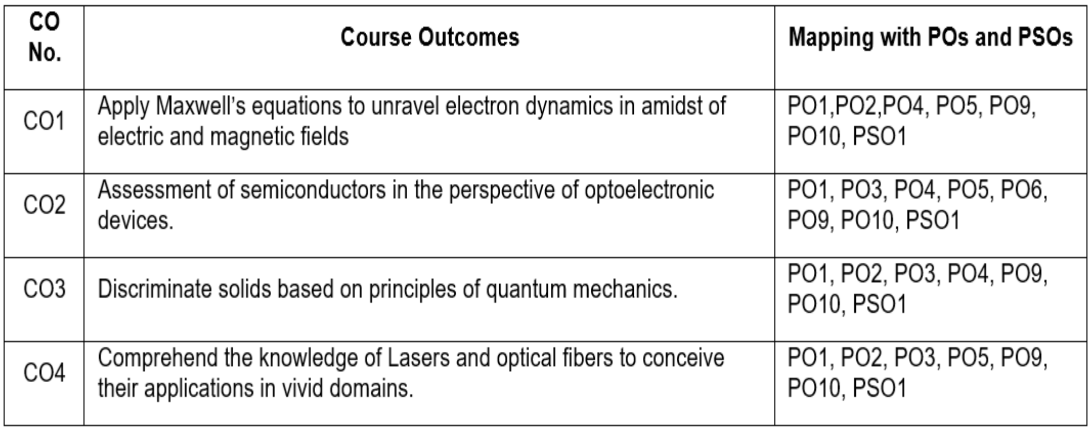
Introduction to lasers, Population inversion & pumping processes, Semiconductor diode laser, Applications of lasers. Optical fiber-Numerical Aperture, types of optical fibres, Fiber optic communication system.

PRACTICES:

- Laser - Determination of wavelength.
- Optical fibre - Determination of Numerical aperture – Acceptance angle.
- Determination of Energy Band gap of p-n junction diode.
- LED - Determination of Threshold Voltage of LED.
- Solar cell – Determination of Fill factor & efficiency.

COURSE OUTCOMES:

Upon successful completion of the course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply Maxwell's equations to unravel electron dynamics in amidst of electric and magnetic fields	PO1,PO2,PO4, PO5, PO9, PO10, PSO1
CO2	Assessment of semiconductors in the perspective of optoelectronic devices.	PO1, PO3, PO4, PO5, PO6, PO9, PO10, PSO1
CO3	Discriminate solids based on principles of quantum mechanics.	PO1, PO2, PO3, PO4, PO9, PO10, PSO1
CO4	Comprehend the knowledge of Lasers and optical fibers to conceive their applications in vivid domains.	PO1, PO2, PO3, PO5, PO9, PO10, PSO1

TEXT BOOKS:

1. S.O. Pillai, “Solid State Physics”, New age International publishers, 8th edition, 2018.
2. H.C. Varma, “Classical Electromagnetism”, Bharathi Bhavan Publication, 2022.

REFERENCEBOOKS:

1. D. Halliday, R. Resnick and J. Walker, “Fundamentals of Physics”, 6th edition, John Wiley and Sons, New York, 2001.
2. M.N. Avadhanulu, “Engineering Physics”, S. Chand publications 2010.
3. Charles Kittel, “Introduction to Solid State Physics”, 7th edition, Wiley, Delhi, 2007.
4. Donald A. Neamen, “Semiconductor Physics and Devices: Basic Principle”, 4th edition, McGraw-Hill, New York, 2012.
5. David J. Griffiths, “Introduction to Electrodynamics”, 3rd edition, Prentice Hall of India, New Delhi, 2012.
6. N.W. Ashcroft and N.D. Mermin, “Solid State Physics”, International student edition, Brooks Cole, 2008.

22EE101 BASICS OF ELECTRICAL AND ELECTRONICS ENGINEERING

Source : https://vita.vision.org.in/emerging-technologies-in-electrical-engineering/

Hours Per Week :

L	T	P	C
2	-	2	3

PREREQUISITE KNOWLEDGE: Electrostatics and Electromagnetism.

COURSE DESCRIPTION AND OBJECTIVES:

This course provides an insight into the functioning of basic electrical components like resistor, inductor and capacitor. It deals with the constructional and operational details of AC machines. It also deals with the basic electronic components like P-N junction diode, Zener diode, Transistor and their characteristics.

MODULE–1

UNIT-1 **8L+0T+8P=16 Hours**

FUNDAMENTALS OF ELECTRIC CIRCUITS:

DC Circuits: Concept of network, Active and passive elements, Voltage and current sources, Concept of linearity and linear network, Unilateral and bilateral elements, R, L and C as linear elements, Ohm's Law, Kirchhoff's Laws, Application to simple series, Parallel circuits, Mesh and nodal analysis of resistive circuits with DC source.

AC circuits: Generation of AC voltage, Frequency, Average value, R.M.S. value, Form factor, Peak factor for sinusoidal only.

UNIT-2 **8L+0T+8P=16 Hours**

SEMICONDUCTOR DEVICES:

Classification of semiconductors, P-N junction diode -operation and its characteristics, Half wave rectifier - operation, efficiency; Full wave rectifiers -types, operation, efficiency; Zener diode and its characteristics, Zener diode as Voltage regulator.

Bi polar junction transistor- operation, types (NPN & PNP).

PRACTICES

- Verification of Ohm's law.
- Verification of Kirchhoff's current law.
- Verification of Kirchhoff's voltage law.
- Determination of R.M.S. Values of sinusoidal waveform.
- Verification of PN junction diode characteristics under both forward and reverse bias.
- Verification of Zener diode characteristics under reverse bias.

MODULE–2

UNIT-1 **8L+0T+8P=16 Hours**

ANALYSIS OF AC CIRCUITS:

Analysis of single- phase ac circuits consisting of R, L, C, RL, RC (series and parallel) (simple numerical problems). Introduction to three phase system, Relation between phase and line quantities of voltages and currents in star and delta connected systems (Elementary treatment only).

SKILLS:

- ✓ *Distinguish between linear and nonlinear elements by looking at VI characteristics.*
- ✓ *Develop a simple loop generator.*
- ✓ *Design a voltage regulator using Zener diode.*
- ✓ *Design a half wave rectifier using PN junction diode.*
- ✓ *Design a full wave rectifier using PN junction diodes.*

UNIT-2 **8L+0T+8P=16 Hours**

AC MACHINES:

Electromagnetism: Concepts of Magneto motive force, Reluctance, Flux and flux density, Concept of self-inductance and mutual inductance, Coefficient of coupling.

Static & Rotating AC Machine: Principle of operation of single phase transformer, Constructional features, EMF equation (simple numerical problems).

Rotating AC Machine Principle of operation of three phase induction motor, Slip ring and squirrel cage motors, Torque equation; Constructional details of synchronous machine.

PRACTICES

- Transformation ratio of a single phase transformer at different loads.
- Measurement of Energy in single phase resistive load circuit.
- Measurement of Power in single phase resistive load circuit
- Determination of impedance in complex AC circuits.
- Verification of line and phase quantities in a balanced three phase system.

COURSE OUTCOMES:

Upon successful completion of the course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Solve the AC (single and three phase) and DC circuits using different methods.	PO1,PO2,PO9,PO12,PSO1
CO2	Apply the concepts of electromagnetism for its applications.	PO1,PO2,PO3,PO9,PO12,PSO1
CO3	Analyze the resistive circuits with independent sources and find its solution.	PO1,PO2,PO6,PO9,PO12,PSO1
CO4	Acquire the knowledge of semiconductor devices to create circuits.	PO1,PO2,PO3,PO9,PO12,PSO1

TEXT BOOKS:

1. V. K. Mehta, “Principles of Electrical Engineering and Electronics”, S.Chand& Co., Publications, New Delhi, 2019.
2. D.P. Kothari, “Basic Electrical and Electronics Engineering”, TMH, New Delhi, 2017.

REFERENCE BOOKS:

1. Millman and Halkias, “Electronic Devices and Circuits”, Mc Graw Hill, 2006.
2. A.K. Thereja and B.L.Thereja, “Electrical Technology”, Vol.–II, S. Chand & Co., Publications, 2020.
3. U. Bakshi and A. Bakshi, “Basic Electrical Engineering”, 1st edition, Technical Publications, Pune, Nov 2020.

22IT102 IT WORKSHOP AND TOOLS

Image source: BookBildr (Created by own from website)

Hours Per Week :

L	T	P	C
-	2	4	3

PREREQUISITE KNOWLEDGE: Computer Knowledge

COURSE DESCRIPTION AND OBJECTIVES:

This course explores the students towards the computer anatomy, office tools like word, excel, power point presentation and basics of computer networks. Make the student to prepare the reports and documents for the assignments, home tests and projects which will be carried out during their academic activities. In addition, it aims to make the student to create and develop documents and power point presentation using latex also.

MODULE-1

UNIT-1 **0L+6T+12P=18 Hours**

INTRODUCTION TO COMPUTER COMPONENT AND COMPUTER NETWORKS:

Computer Components and MS Office: Introduction to Computers, Basic Components of PC, Network Essentials, System Security, Networking basics, Uses of network, Types of computer networks, Network hardware.

UNIT-2 **0L+10T+20P=30 Hours**

COMPUTER NETWORKS AND MS-OFFICE:

Computer Networks: Layers of networks, Devices of computer networks.

MS Word: Design Certificate and Resume, Creating a Newsletter, Preparation of a single column Report, IEEE format and Springer format.

MS Excel: Prepare invoice and purchase order for the M & M Pvt. Ltd., Calculating GPA marksheet report.

MS PowerPoint: Prepare a PowerPoint template for department facilities, Create a Business Agenda Slide in a PowerPoint presentation, Create Animation and Transitions based company profile presentation slide.

PRACTICES:

- Create a network using different devices.
- Design a certificate for cricket sport.
- Creation of projects and Newsletter using MS Word.
- Create a single column document for any topic regrading concert subject.
- Prepare an IEEE format using following.
  - Insert a multiple types of tables
  - Insert a Figures
  - Insert an Equations
- Prepare a Springer format using following.
  - Insert a horizontal and vertical tables
  - Insert multiple figures in column wise
  - Insert multiple figures in a row wise
  - Insert multiple figures in both row and column wise
  - Insert an Equations

SKILLS:

- ✓ *Basics of computer components and computer networks.*
- ✓ *Installation of operating system and secure the system.*
- ✓ *Hands on text processing and typesetting software, simple data analysis and data presentation tools.*
- ✓ *Create reports and article in proper format*

- Spreadsheet basics, modifying worksheets, formatting cells, formulas and functions, sorting and filtering, charts using MS Excel.
- Prepare invoice and purchase order for the M & M Pvt. Ltd.
- Calculating CGPA mark sheet report in bar chart, pie chart.
- Create PowerPoint presentations for any concept using power point screen, working with slides, add content, work with text, working with tables, graphics, slide animation, reordering slides, adding sound to a presentation using MS PPT.

MODULE-2

UNIT-1 **0L+6T+12P=18 Hours**

LATEX:

Introduction to Latex: Introduction, Scope of Latex, Overview of Overleaf: Different types of the files.

Latex Installation and Demonstration: Texstudio and Textlive/Miktex.

UNIT-2 **0L+10T+20P=30 Hours**

LATEX DOCUMENT PREPARATION:

Preparation of document structure, create mathematics equation, add tables, figures; header and footer, bibliography for article, creating a newsletter, preparation of a single column report, IEEE format, springer format and Presentation with beamer package.

PRACTICES:

- Install TexLive software and supporting files.
- Create sample document in Overleaf.
- Create sample document file in in Latex desktop software.
- Creation of projects and Newsletter using LaTeX.
- Create a single column document for any topic regrading concert subject in latex.
- Prepare an IEEE format using following in latex.
  - Insert a multiple types of tables
  - Insert a Figures
  - Insert an Equations
- Prepare a Springer format using following in latex.
  - Insert a horizontal and vertical tables
  - Insert multiple figures in column wise
  - Insert multiple figures in a row wise
  - Insert multiple figures in both row and column wise
  - Insert an equation
- Prepare a Springer format using following in latex.
  - Insert references
  - Insert multiple figures in column wise
  - Insert multiple figures in a row wise
  - Insert multiple figures in both row and column wise
  - Insert an equation
- Create PowerPoint for any topic using Latex.

COURSE OUTCOMES:

Upon successful completion of the course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply basic knowledge of computer components, networking concepts, and security practices to operate and manage a personal computer in a connected environment.	PO1,PO2,PO3,PO5,PO9,PO10, PO12, PSO1, PSO2, PSO3
CO2	Analyze layered network models and hardware devices to explain how data is transmitted and managed in computer networks.	PO1,PO2,PO3,PO5,PO9,PO10, PO12, PSO1, PSO2, PSO3
CO3	Evaluate the layout, structure, and formatting of documents and presentations created using MS Office and LaTeX based on professional and academic standards.	PO1,PO2,PO3,PO5,PO9,PO10, PO12, PSO1, PSO2, PSO3
CO4	Create well-structured academic, technical, and business documents using MS Office and LaTeX tools, incorporating appropriate templates, formatting, and content.	PO1,PO2,PO3,PO5,PO9,PO10, PO12, PSO1, PSO2, PSO3

TEXT BOOKS:

1. Priti Sinha and Pradeep K. Sinha, “Computer Fundamentals: Concepts, Systems and Applications”, 8th Edition, BPB Publications, 2004
2. Stefan Kottwitz, “LaTeX Beginner’s Guide: Create visually appealing texts, articles, and books for business and science using LaTeX”, 2nd Edition, Kindle, 2021.

REFERENCE BOOKS:

1. Michael R. Miller,” Computer Basics Absolute Beginner’s Guide, Windows 8 Edition”, 6th Edition, Kindle Edition,2012.
2. Satish Jain, Shalini Jain and M.Geetha, “ Basic Computer Course Made Simple” PBS Publications, 2016.
3. John Monyjok Maluth, “Basic Computer Knowledge”, 5th Edition, Kindle, 2016.
4. Antoni Diller, “LaTeX: Line by Line: Tips and Techniques for Document Processing”, 2nd Edition, Wiley, 1998.

✓ Techgig.com

22TP103 PROGRAMMING IN C

Hours Per Week :

L	T	P	C
2	-	4	4

PREREQUISITE KNOWLEDGE: Fundamentals of Problem Solving.

COURSE DESCRIPTION AND OBJECTIVES:

This course is aimed to impart knowledge on basic concepts of C programming language and problem solving through programming. It covers basic structure of C program, data types, operators, decision making statements, loops, functions, strings, pointers, and also file manipulations. At the end of this course, students will be able to design, implement, test and debug complex problems using features of C.

MODULE- 1

UNIT-1 **8L+0T+16P=24 Hours**

INTRODUCTION TO ALGORITHMS AND PROGRAMMING LANGUAGES:

Introduction to Algorithms: Basics of algorithms; Flow charts; Generations of programming languages. Introduction to C: Structure of a C program - pre-processor statement, inline comments, variable declaration statements, executable statements; C Tokens - C character set, identifiers and keywords, type qualifiers, type modifiers, variables, constants, punctuations and operators.

Data Types and Operators: Basic data types; Storage classes; Scope of a variable; Formatted I/O; Reading and writing characters; Operators - assignment, arithmetic, relational, logical, bitwise, ternary, address, indirection, sizeof, dot, arrow, parentheses operators; Expressions - operator precedence, associative rules.

Control Statements: Introduction to category of control statements; Conditional branching statements - if, if– else, nested-if, if – else ladder, switch case; Iterative statements - for, while, do - while, nested loops; Jump statements - break, jump, goto and continue.

UNIT-2 **8L+0T+16P=24 Hours**

ARRAYS & STRINGS:

Arrays: Introduction; Types of arrays; Single dimensional array - declaration, initialization, usage, reading, writing, accessing, memory representation, operations; Multidimensional arrays.

Strings: Character array, Reading string from the standard input device, Displaying strings on the standard output device, Importance of terminating a string, Standard string library functions.

PRACTICES:

Questions on Data Handling – Level 1:

- Write a program to accept a character as input from the user and print it.
- Write a program to accept a number as input from the user and print it.
- Write a program to accept a float value from the user and print it.
- Write a program to accept a message as input from the user and print it.
- Write a program to accept a message from the user as input and print it in 3 different lines.
- Write a program to accept 2 numbers from the user as input and print their sum.
- Write a program to accept 2 numbers from the user as input and print their product.
- Write a program to accept a number as input from the user which denotes the temperature in Celsius, convert it to Fahrenheit reading and print it.

- Write a program to accept a number as input from the user which denotes the radius and print the area of the circle.
- Write a program to accept a character as input from the user and print it’s corresponding ASCII value.

Questions on Control Statements - Looping – Level 1:

- Write a C program to print all the characters from a to z once.
- Write a C program to print all the characters from Z to A once.
- Write a C program to print all the characters from A to Z 3 times.
- Write a C program to print the first N natural numbers, where N is given as input by the user.
- Write a C program to print the first N natural numbers and their sum, where N is given as input by the user.
- Write a C program to print all the odd numbers between 1 and N where N is given as input by the user.
- Write a C program to print all the even numbers between I and N where N is given as input by the user.
- Write a C program to print the squares of the first N natural numbers between 1 and N, where N is given as input by the user.
- Write a C program to print the cubes of the first N natural numbers between 1 and N, where N is given as input by the user.
- Write a C program to print the squares of every 5th number starting from 1 to N, where N is given as input by the user.

Questions on Control Statements – Decision Making – Level 1:

- Write a program to accept two numbers as input check if they are equal.
- Write a program to accept two characters as input and check if they are equal.
- Write a program to accept two numbers as input and print the greater of the 2 numbers.
- Write a program to accept two numbers as input and print the lesser of the 2 numbers.
- Write a program to accept 3 numbers as input and print the maximum of the 3.
- Write a program to accept 3 numbers as input and print the minimum of the 3.
- Write a program to accept a number as input and print EVEN if it is an even number and ODD if it is an odd number.
- Write a program to accept a number as input and check if it is divisible by 3. If it is divisible by 3 print YES else print NO.
- Write a program to accept a number as input and check if it is divisible by both 3 & 5. If it is divisible print YES else print NO.
- Write a program to accept a number as input and check if it is positive, negative or zero.

Questions on Patterns – Level 1:

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  ```
  *****
  *****
  *****
  *****
  *****
  ```

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  ```
  *****
  *   *
  *   *
  *   *
  *****
  ```

SKILLS:

- ✓ *Analysis of the problem to be solved.*
- ✓ *Select static or dynamic data structures for a given problem and manipulation of data items.*
- ✓ *Application of various file operations effectively in solving real world problems.*
- ✓ *Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.*

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  ```
  *
  **
  ***
  ****
  *****
  ```

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  ```
      *
     **
    ***
   ****
  *****
  ```

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  ```
  1
  12
  123
  1234
  12345
  ```

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  ```
  1
  22
  333
  4444
  55555
  ```

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  ```
  54321
  4321
  321
  21
  1
  ```

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  ```
  12345
  2345
  345
  45
  5
  ```

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  ```
  A
  AB
  ABC
  ABCD
  ABCDE
  ```

- Write a program to accept a number N as input from the user and print the following pattern. Sample N = 5.

  A
  BC
  DEF
  GHIJ
  KLMNO

Questions on Number Crunching – Level 1:

- Write a program to accept a number as input and print the number of digits in the number.
- Write a program to accept a number as input print the sum of its digits.
- Write a program to accept a number as input, reverse the number and print it.
- Write a program to accept a number and digit as input and find the number of occurrences of the digit in the number.
- Write a program to accept a number as input and check if it is an Armstrong number.
- Write a program to accept a number as input and check if it is an Adam number.
- Write a program to accept a number as input and check if is a prime number.
- Write a program to accept 2 numbers as input and check if they are amicable or not.
- Write a program to accept a number as input and check if it is a power of 2.
- Write a program to accept 2 numbers as input and find their LCM.

Questions on Arrays – Level 1:

- Print the contents of an array from the left to the right.
- Print the contents of an array from the right to the left.
- Find the sum of the elements of an array.
- Find the maximum element in an unsorted array.
- Find the minimum element in an unsorted array.
- Find the average of the elements in an unsorted array.
- Count the number of 0s and 1s in an array having 0s and 1s in random order.
- Count the number of elements in an array whose elements are lesser than a key element in an unsorted array.
- Print all the elements in an array whose values are lesser than a key element in an unsorted array.
- Find the repeated elements in a sorted array.

Questions Number crunching – Level 2:

- Write a program to accept a number as input and print the product of its digits.
- Write a program to accept a number as input and check if it is a palindrome.
- Write a program to accept a number as input and print the frequency of occurrence of each digit.
- Write a program to accept a number as input and print its factors.
- Write a program to accept a number as input and print its prime factors.
- Write a program to accept a number as input and check if it is a perfect square of not.
- Write a program to accept 2 numbers as input and check if they are betrothed numbers or not.
- Write a program to accept 2 numbers as input and print their HCF.
- Write a program to accept a number as input and check if is a strong number.
- Write a program to generate prime numbers between two intervals given as input.

Questions on Arrays – Level 2:

- Find the sum of the maximum and minimum numbers of an unsorted array.
- Replace every element in an array with the sum of its every other element.
- Replace every element in an array with the sum of its right side elements.
- Replace every element in an array with the sum of its left side elements.
- Reverse the elements of an array (in place replacement).
- Reverse the first half of an array.

- Reverse the second half of an array.
- Write a program to find the second largest element in an unsorted array.
- Write a program to find the second smallest element in an unsorted array.
- Write a program to print the number of odd and even numbers in an unsorted array.

Questions on Strings – Level 1:

- Write a program to accept a string as input and print it.
- Write a program to accept a string as input and count the number of vowels in it.
- Write a program to accept a string as input and count the number of consonants in it.
- Write a program to accept a string as input and print its length.
- Write a program to accept a string as input and print the reversed string.
- Write a program to accept 2 strings as input and check if they are the same.
- Write a program to accept a string as input and copy the contents into a second string and print the second string.
- Write a program to accept 2 strings as input and concatenate them into a third string and print the third string.
- Write a program to accept a string as input and check if it is a palindrome.
- Write a program to accept two strings as input and check if the second string is a substring of the first.

Questions on Strings – Level 2:

- Implement the string length function.
- Implement the string copy function.
- Implement the string concatenate function.
- Implement the string compare function.
- Implement the vowel count function.
- Implement the consonant count function.
- Implement the count words function.
- Implement the string reverse function.
- Implement the strstr function.
- Complete the code snippet to implement the is Palindrome function that checks if a given string is a palindrome. You will need to use the 3 functions string Copy, str Reverse and string Compare functions provided to accomplish this.

MODULE-2

UNIT-1 **8L+0T+16P=24 Hours**

FUNCTIONS & POINTERS:

User-defined functions: Function declaration - definition, header of a function, body of a function, function invocation; Call by value; Call by address; Passing arrays to functions; Command line arguments; Recursion; Library Functions.

Pointers: Declaration, Initialization, Multiple indirection, Pointer arithmetic, Relationship between arrays and pointers, Scaling up - array of arrays, array of pointers, pointer to a pointer and pointer to an array; Dynamic memory allocation functions.

UNIT- 2 **8L+0T+16P=24 Hours**

STRUCTURES, UNIONS & FILES:

Structures: Defining a structure, Declaring structure variable, Operations on structures, Pointers to structure - declaring pointer to a structure, accessing structure members using pointer; Array of structures, Nested structures, Passing structures to functions - passing each member of a structure as a separate argument, passing structure variable by value, passing structure variable by reference/ address; Typedef and structures.

Unions: Defining a union - declaring union variable, operations on union; Pointers to union - declaring pointer to a union, accessing union members using pointer; Array of union, Nested union, Typedef and union, Enumerations, Bit-fields.

Files: Introduction to files, Streams, I/O using streams – opening a stream, closing stream; Character input, Character output, File position indicator, End of file and errors, Line input and line output, Formatted I/O, Block input and output, File type, Files and command line arguments.

PRACTICES:

Questions on Strings – Level 3:

- Write a program to swap two given strings and print the swapped strings.
- Write a program to swap two given words of the given sentence and print the altered string.
- Return the maximum occurring character in the string.
- Write a program to print the character in the string with the count where count is the occurrence of the character.
- Write a program to print the duplicate characters in the given string.
- Write a program to remove the duplicate characters in the given string.
- Write a program to remove the vowels from a given string.
- Write a program to rotate a given string N number of times.
- Write a program to check if 2 strings are rotations of each other.
- Write a program to remove the characters from the first string that are present in the second string.

Questions on 2D Arrays – Level 1:

- Print the contents of a 2D array row-wise.
- Print the contents of a 2D array column-wise.
- Print the contents of a 2D array in a zig-zag order.
- Print the contents of a 2D array diagonal-wise.
- Print the contents of a 2D array right-diagonal order.
- Print the contents of a 2D array left-diagonal order.
- Print the contents of a 2D array in the upper triangular order – left top to right bottom.
- Print the contents of a 2D array in the lower triangular order.
- Find and print the maximum element along with its position in a matrix.
- Find and print the minimum element along with its position in a matrix.

Questions on 2D Arrays – Level 2:

- Find and print the maximum element of each row of a matrix.
- Find and print the minimum elements of each row of a matrix.
- Find and print the maximum element of each column of a matrix.
- Find and print the minimum element of each column of a matrix.
- Find the lowest value in the upper triangle area and the largest value in the lower triangular area of a matrix and print their product.
- Find the sum of the elements of each row and each column of a matrix and print the minimum row sum and maximum sum column.
- Write a program to find the row with the maximum number of 1's in a matrix consisting of only 0's and 1's.
- Write a program to print the quotient and remainder on dividing sum of left-top to right-bottom diagonal by sum of right-top to left-bottom diagonal.
- Write a program to print the absolute difference of the sum of major diagonal elements and the sum of minor diagonals of the given matrix.
- Write a program to search a given element in a row-wise and column-wise sorted 2D array.

Questions on 2D Arrays – Level 3:

- Write a program to find the Kth smallest element in the given matrix.
- Write a program to find the Kth largest element in the given matrix.

- Write a program to check whether the given two two-dimensional array of same dimensions are equal or not.
- Write a program to add the given two two-dimensional array of same dimensions.
- Write a program to subtract the given two two-dimensional array of same dimensions.
- Write a program to multiply the given two two-dimensional array of same dimensions.
- Write a program to sort each row of a matrix.
- Write a program to find the sum of the elements in 'Z' sequence of the given 2D array.
- Write a program to print the unique rows of the given two-dimensional array consisting of only 0's and 1's.
- Write a program to print the unique columns of the given two-dimensional array consisting of only 0's and 1's.

Questions on Files, Structures & Unions:

- Write a C program to create a struct, named Student, representing the student's details as follows: first_name, last_name, Age and standard.

  Example

  Read student data

  john

  carmack

  15

  10

  Display the data in the following format

  First Name: john

  Last Name: carmack

  Age: 15

  Standard: 10

- Declare a structure POINT. Input the coordinates of point variable and write a C program to determine the quadrant in which it lies. The following table can be used to determine the quadrant.

Quadrant	**X**	**Y**
1	Positive	Positive
2	Negative	Positive
3	Negative	Negative
4	Positive	Negative

  Example

  Input the values for X and Y coordinate: 7 9

  The coordinate point (7,9) lies in the First quadrant.

- Bob and Alice both are friends. Bob asked Alice how to store the information of the books using Structures. Then Alice written a c program to store the information of books using book structure by taking different attributes like book_name, author, book_id, price. Write a C program to read and display the attributes of the books using structures.

  Sample Input:

  Enter number of books: 1

  Enter the book name: c Programming

  Enter the author name: balaguruswamy

  Enter the book ID: 23413

  Enter the book price: 500

  Sample Output:

  The details of the book are:

  The book name is: c Programming

The author name is: balaguruswamy

The book ID is: 23413

The book price is: 500.00

- Ramesh wants to do addition on complex numbers. He did it with regular practice but Charan asked him to do with the help of structures by following below Criteria.

  Write a C program that defines a structure named ‘Complex’ consisting of two floating point members called “real and imaginary”. Let c1 and c2 are two Complex variables; compute the sum of two variables.

  Example:

  c1= 2 8
  c2= 6 4
  Sum= 8.000000+12.000000i

- Customer Payment Details is a structure with members as customers_name, address, account_number, payment_status(paid(1)/ not_paid(0)), due_date, and amount. In this example, payment_date is another structure with month, day and year as integer members. So, every customer record can be considered as an array of structures.

  Write a C program that displays the amount to be paid by each customer along with their names. If payment_status is 1, display NIL for such customers.

  Input Format:

  First line of input contains ‘n’ number of customers, followed by 8 lines of input for each customer. Each line represents (customers_name, address, account_number, amount payment_status(paid(1)/ not_paid(0)), and due_date).

  Output Format:

  First line of output is Amount to be paid by each customer as on date: followed by n lines of output. Each line contains name of the customer followed by tab space, and amount to be paid.

  Hint: Use nested structure to represent date.

  Write a ‘C’ program to accept customer details such as: Account_no, Name, Balance using structure. Assume 3 customers in the bank. Write a function to print the account no. and name of each customer whose balance < 100 Rs.

- Write a C program to accept details of ‘n’ employee(eno, ename, salary) and display the details of employee having highest salary. Use array of structure.

- Write a C program to print the bill details of ‘N’ number of customers with the following data: meter number, customer name, no of units consumed, bill date, last date to deposit and city. The bill is to be calculated according to the following conditions:

No. of units	**Charges**
For first 100 units	Rs.0.75 per unit
For the next 200 units	Rs.1.80 per unit
For the next 200 units	Rs.2.75 per unit

Sample Input

Enter no. of customers

1

Enter Meter Number AP01213

Enter Customer Name: Karthik

Enter No. of units consumed: 200

Enter Bill date:22/01/2021

Enter Last date: 12/2/2021

Enter City: Guntur

Sample Output

Meter Number AP01213

Customer Name: Karthik

No. of units consumed: 200

Bill date:22/01/2021

Last date: 12/2/2021

City: Guntur

Total Amount: 255.000000

- Write a C program that creates a student file containing {Roll No, Student Name, Address, Stream}, where the data will be inserted and display the list of students who are in CSE (Stream=CSE).

  Input: A file name

  Output: The attributes such as Roll_No, Student_Name, Stream, Address.

  Sample Input

  ```
  201fa4200     Raja    CSE    Guntur
  201fa4201     Bala    IT     Tenali
  ```

  Sample Output

  ```
  201fa4200     Raja    CSE    Guntur
  ```

- Write a C program that reads content from an existing text file and write the same in a new file by changing all lowercase alphabetic character to upper case. (Existing file may contain digit and special characters).

  Example:

  Input: Enter the file name.

  Output: New file with updated content.

- Write a C program to count the occurrences of the given string in a file.

  Example:

  Input: Enter the File name to read the string to be counted.

  Output: Display the count of occurrences of the string.

- Write a C Program to transfer the data from one location to another location without changing the order of the content.

  Example:

  Read the file name from the user. If the source file exists, Transfer the data and display the message as “Data is transferred successfully” otherwise display the message “No such file is existing in the directory.”

- Write a C program that reads numbers and write them into a text-file. Also find odd and even numbers in that file and store it in 2 separate files named odd.txt and even.txt. All the values should be in ascending order.

  Input: Enter the values.

  Output: Creates a separate file for Even and Odd numbers.

  Sample Input:

  4 43 2 53 45

  Sample Output:

  Even.txt: 2 4

  Odd.txt: 43 45 53

- Write a C program to replace the content in the given text file.

  Input: Enter the file name, line number to be replaced and the new content

  Output: New file with replaced lines.

  Example:

  Sample Input: Enter the file name: abc.txt

  Enter the line no to replace: 3

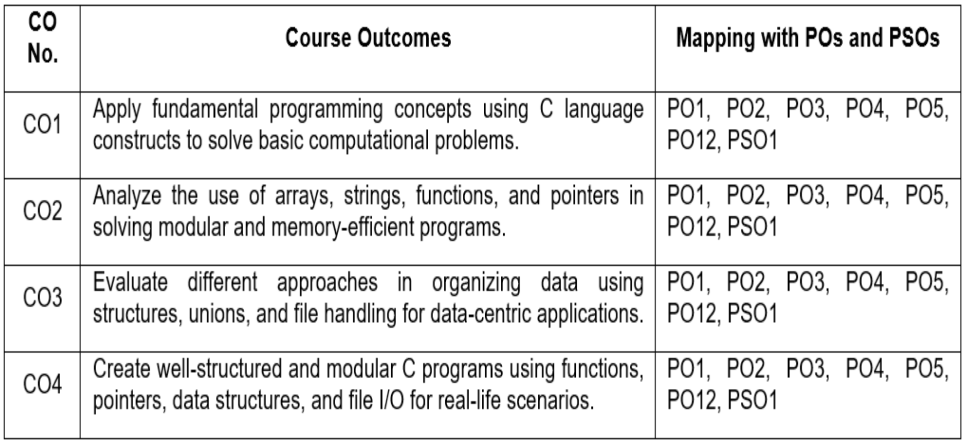
  Enter the content: Files stores data presently.

Sample Output:

```
Line no 3 is replaced with the given content.
The content of the file abc.txt contains:
test line 1
test line 2
Files stores data presently
test line 4
```

COURSE OUTCOMES:

Upon successful completion of the course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply fundamental programming concepts using C language constructs to solve basic computational problems.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1
CO2	Analyze the use of arrays, strings, functions, and pointers in solving modular and memory-efficient programs.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1
CO3	Evaluate different approaches in organizing data using structures, unions, and file handling for data-centric applications.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1
CO4	Create well-structured and modular C programs using functions, pointers, data structures, and file I/O for real-life scenarios.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1

TEXT BOOKS:

1. Behrouz A. Forouzan, Richard F.Gilberg, “Programming for Problem Solving”, 1st edition, Cengage publications, 2019.
2. Ajay Mittal, “Programming in C - A Practical Approach”, 1st edition, Pearson Education, India, 2010.

REFERENCE BOOKS:

1. Reema Thareja, “Computer Fundamentals and Programming in C”, 1st edition, Oxford University Press, India, 2013.
2. Herbert Schildt, “C: The Complete Reference”, 4th edition, Tata McGraw-Hill, 2017.
3. Byron S Gottfried, “Programming with C”, 4th edition, Tata McGraw-Hill, 2018.

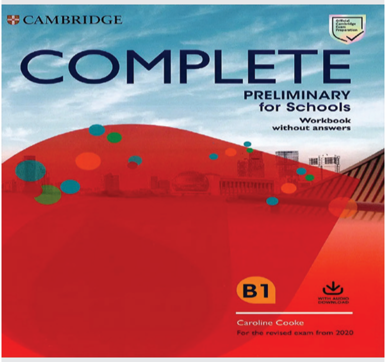


Image source: https://www.scribd.com/document/502301821/Cambridge-Complete-B1-Preliminary-for-Schools-Workbook-2020-Edition

22EN102 ENGLISH PROFICIENCY AND COMMUNICATION SKILLS

Hours Per Week :

L	T	P	C
-	-	2	1

PREREQUISITE KNOWLEDGE: Basics of grammar, Read and understand for global context, Cultural sensitivity and Basic writing skills.

COURSE DESCRIPTION AND OBJECTIVES:

English Proficiency and Communication Skills seeks to develop the students' abilities in grammar, speaking, reading, writing and overall comprehension skills. The course will provide students an exposure on a wide range of language use in everyday situations. It will make the students to equip with functional English and make them use it confidently in their professional and social contexts. Finally, students will strengthen their reading, writing, listening and speaking skills in English

MODULE-1

UNIT-1 **0L+0T+8P=8 Hours**

MY LIFE AND HOME – MAKING CHOICES – HAVING FUN:

Reading: Understanding main message, factual information global meaning, specific information and paraphrasing.

Writing: Developing hints based mail, Writing short messages/paragraphs.

Listening: Understanding short monologues or dialogues and choose the correct visual.

Speaking: Express simple opinions /cultural matters in a limited way.

Vocabulary: Discerning use of right word suiting the context, B1 Preliminary word list.

Grammar: Frequency Adverbs, State Verbs, AFV and Prepositions.

UNIT-2 **0L+0T+8P=8 Hours**

ON HOLIDAY - DIFFERENT FEELINGS – THAT'S ENTERTAINMENT!:

Reading: Longer text for detailed comprehension, gist and inference.

Writing: Developing notes and responding to penfriends or 'e-pals'.

Listening: Understand straightforward instructions or public announcements.

Speaking: Describing people, things and places in a photograph.

Vocabulary/Grammar:

Comparatives and Superlatives, Gradable and non-gradable adjectives, Cloze tests.

PRACTICES:

- Developing hints based mail.
- Writing short message.
- Writing paragraphs.
- Expressing opinions and cultural matters.
- Understanding short monologues.
- Understanding straightforward instructions and public announcements.
- Describing people, things and places in a photograph.

MODULE-2

UNIT-1 **0L+0T+8P=8 Hours**

GETTING AROUND – INFLUENCES - STAY FIT AND HEALTHY:

Reading: Reading for understanding coherence of the text and drawing inferences.

Writing: Reading an announcement from a magazine or website for preparing an article.

Listening: Discussion activities and listening to understand the gist of each short dialogue.

Speaking: Snap Talks, Make and respond to suggestions, discuss alternatives and negotiate agreement.

Vocabulary / Grammar: Punctuation, Prepositions, Phrasal Verbs, B1 Preliminary word list.

UNIT-2 **0L+0T+8P=8 Hours**

LOOKS AMAZING! – THE NATURAL WORLD – EXPRESS YOURSELF!:

Reading: Content, Communicative Achievement, Organisation and Language.

Writing: Developing a story with clear links to the given opening sentence.

Listening: An interview for a detailed understanding of meaning and to identify attitudes and opinions.

Speaking: Discuss likes, dislikes, experiences, opinions, habits, etc.

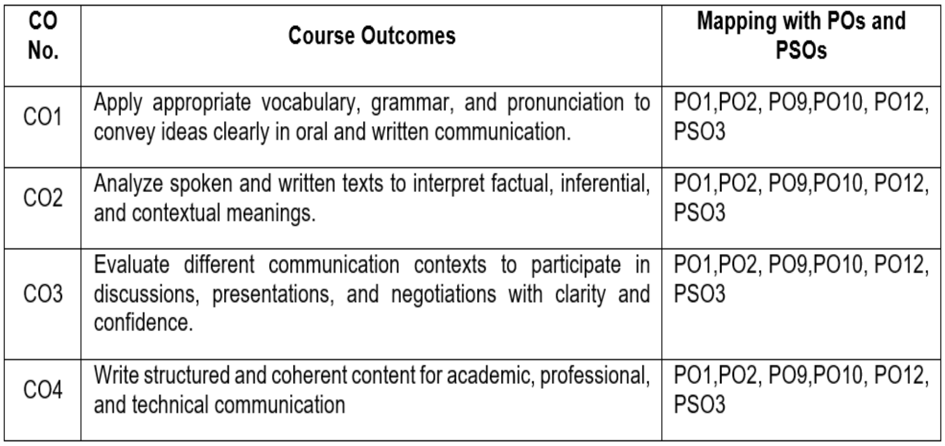
Vocabulary/Grammar: Modals, Conditionals, Verb forms (Time and Tense).

PRACTICES:

- Listening to understand the gist of each short dialogue.
- Listening to an interview for a detailed understanding of meaning and to identify attitudes and opinions.
- Preparing an article.
- Discuss for alternatives and negotiate agreement.
- Discussion on likes, dislikes, experiences, opinions, habits, etc.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply appropriate vocabulary, grammar, and pronunciation to convey ideas clearly in oral and written communication.	PO1,PO2, PO9,PO10, PO12, PSO3
CO2	Analyze spoken and written texts to interpret factual, inferential, and contextual meanings.	PO1,PO2, PO9,PO10, PO12, PSO3
CO3	Evaluate different communication contexts to participate in discussions, presentations, and negotiations with clarity and confidence.	PO1,PO2, PO9,PO10, PO12, PSO3
CO4	Write structured and coherent content for academic, professional, and technical communication	PO1,PO2, PO9,PO10, PO12, PSO3

TEXT BOOKS:

1. Emma Heyderman and Peter May, “Complete Preliminary”, Student’s Book with Answers, 2nd edition, Cambridge University Press, 2019.

REFERENCE BOOKS:

1. Annette Capel and Rosemary Nixon, “Introduction to PET”, Oxford University Press, 2009.
2. Adrian Doff and Craig Thaine, “Empower Pre intermediate”, Cambridge University Press, 2015.
3. Louise Hashemi and Barbara Thomas, “Objective PET”, Cambridge University Press, 2010.

SKILLS:

- ✓ *Use of appropriate grammar and vocabulary with syntactic patterns in short texts.*
- ✓ *Read and extract the main message, global meaning, specific information, detailed comprehension, understanding of attitude, opinion and writer purpose and inference.*
- ✓ *Listen to understand key information, specific information, gist and detailed meaning and to interpret meaning.*
- ✓ *Understand questions and make appropriate responses and talk freely on everyday topics.*

Image: https://commons.wikimedia.org/wiki/File:Constitution_india.jpg

22TP101 CONSTITUTION OF INDIA

Hours Per Week :

L	T	P	C
-	2	-	1

PREREQUISITE KNOWLEDGE: High School-level Civics and Social Studies.

COURSE DESCRIPTION AND OBJECTIVES:

To provide students with a basic understanding of Indian Polity and Constitution and make students understand the functioning of government at the center and state level besides local self-government. This course also equips students with knowledge pertaining to fundamental rights and fundamental duties of a citizen in a democracy such as India.

MODULE-1

UNIT-1 **0L+8T+0P=8 Hours**

HISTORICAL BACKGROUND TO THE INDIAN CONSTITUTION:

Meaning of the constitution law and constitutionalism; Historical perspective of the Constitution of India; Salient features and characteristics of the Constitution of India.

UNIT-2 **0L+8T+0P=8 Hours**

FUNDAMENTAL RIGHTS, DUTIES, DIRECTIVE PRINCIPLES, AND AMENDMENT:

Scheme of the fundamental rights - scheme of the Fundamental Right to Equality; scheme of the Fundamental Right to certain Freedom under Article 19; scope of the Right to Life and Personal Liberty under Article 21; Scheme of the Fundamental Duties and its legal status; Directive Principles of State Policy – its importance and implementation; Amendment of the Constitution - Powers and Procedure.

PRACTICES:

- Enactment of Constituent Assembly debates to further understand the rationale for the provisions of the constitution.
- Fundamental Rights in our popular culture - discussion in the movie Jai Bhim.

MODULE-2

UNIT-1 **0L+8T+0P=8 Hours**

STRUCTURE AND FORM OF GOVERNMENT:

Federal structure and distribution of legislative and financial powers between the Union and the States; Parliamentary Form of Government in India – The constitution powers and status of the President of India; Emergency Provisions: National Emergency, President Rule, Financial Emergency.

UNIT-2 **0L+8T+0P=8 Hours**

LOCAL SELF GOVERNMENT:

Local Self Government – Constitutional Scheme in India – 73rd and 74th Amendments.

PRACTICES:

- Debate on federalism in India.
- Collect news published in the local papers about panchayats in the nearby areas.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the concepts of constitutionalism and analyze the historical background and core features of the Indian Constitution.	PO6, PO8, PO10, PO12, PSO2
CO2	Analyze the structure of fundamental rights, duties, and directive principles to understand their interrelationship and impact on citizens and governance.	PO6, PO8, PO10, PO12, PSO2
CO3	Evaluate the distribution of powers in India's federal structure and assess the role of the President and Emergency provisions in the governance framework.	PO6, PO7, PO8, PO10, PO12, PSO2
CO4	Create a framework for understanding the constitutional role and relevance of local self-government institutions under the 73rd and 74th Amendments.	PO6, PO7, PO8, PO10, PO12, PSO2

TEXTBOOK:

1. PM Bhakshi, “Constitution of India”, 15th edition, Universal Law Publishing, 2018.

REFERENCE BOOKS:

1. B. R. Ambedkar, “The Constitution of India” Educreation Publishing, India, 2020.
2. Subhash Kashyap, “Our Constitution” 2nd edition, National Book Trust, India, 2011.
3. Arun K. Thiruvengadam, “The Constitution of India: A Contextual Analysis”, Hart Publishing India, 2017.

SKILLS:

- ✓ *Understanding the basics of the Indian constitution.*
- ✓ *Know the fundamental rights, fundamental duties, and Directive Principles of State Policy.*
- ✓ *Fair knowledge about the functioning of various institutions in a democracy.*

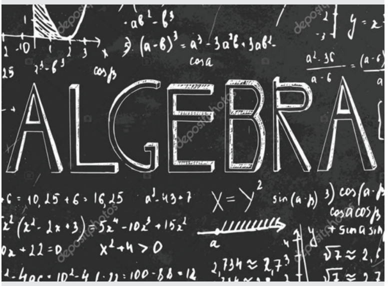


Image Source: https://pll.harvard.edu/course/college-algebra

22MT106 ALGEBRA

Hours Per Week :

L	T	P	C
3	2	-	4

PREREQUISITE KNOWLEDGE: Basics of sets, Relations and Functions.

COURSE DESCRIPTION AND OBJECTIVES:

This course emphasizes on motivation and justification for the algorithmic usage of group theory in different domains. The objective of this course is to introduce the concepts of Groups, Rings, Integral domains and Fields. Develop the ability to form and evaluate group theory and its actions. Understand the fundamental concepts of algebra. The fundamental notions viz. linear dependence, basis and dimension and linear transformations on these spaces have to be studied thoroughly.

MODULE-1

UNIT-1 **12L+8T+0P=20 Hours**

GROUP THEORY:

Algebraic structures with binary operations, Semigroup, Monoid, Group, Subgroup, Cosets, Lagrange's theorem, Normal subgroup, Quotient group.

UNIT-2 **12L+8T+0P=20 Hours**

APPLICATIONS OF GROUP THEORY:

Introduction to Rings, Integral Domains, Fields with examples.

Properties of groups, order of an element in a group, homomorphism, isomorphism.

PRACTICES:

- List all the properties for group
- Give examples for groups and other binary structures.
- In a group of even order there is an element a e such that a2 = e.
- For any two subgroups discuss the possibility of their intersection and union being a subgroup.
- Any two groups of order 6 are isomorphic, Verify.

MODULE-2

UNIT-1 **12L+8T+0P=20 Hours**

VECTOR SPACES:

Vector space, Subspace, linear span, linearly independent and dependent vectors, Bases, Dimension, Linear transformations, Inner product spaces.

UNIT-2 **12L+8T+0P=20 Hours**

APPLICATIONS OF VECTOR SPACES:

Matrix of Linear Transformation, Change of Coordinates, Rank and Nullity, Orthogonality, Cauchy's Schwartz Inequality, Gram Schmidt Orthogonalization.

PRACTICES:

- Examine whether or not a given algebraic structure is a vectorspace.
- Verify whether a given set forms a basis or not of R3.
- Testing orthogonality of given set of vectors.
- Finding Rank and Nullity of linear transformation.

COURSEOUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the concepts of cosets to study properties of subgroups.	PO1, PO2, PO9, PO10, PO12, PSO1
CO2	Outline the various properties and apply group actions critically.	PO1, PO2, PO9, PO10, PO12, PSO1
CO3	Understand and apply the concepts of vector spaces, subspaces, bases, dimension and their properties	PO1, PO2, PO9, PO10, PO12, PSO1
CO4	Analyze inner product spaces for their orthogonality.	PO1, PO2, PO9, PO10, PO12, PSO1

TEXT BOOKS:

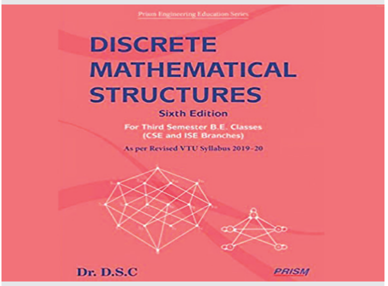
1. Tremblay, J.P. and Manohar. R, “Discrete Mathematical Structures with Applications to Computer Science”, 30th Reprint, Tata McGraw Hill Pub. Co. Ltd, New Delhi, 2017.
2. Rosen, K.H., “Discrete Mathematics and its Applications”, 7th Edition, Tata McGraw Hill Pub. Co. Ltd., New Delhi, 2018.

REFERENCE BOOKS:

1. R.P. Grimaldi, “Discrete and Combinatorial Mathematics: An Applied Introduction”, 4th Edition, Pearson Education Asia, Delhi, 2017.
2. S. Lipschutz and Mark Lipson., “Discrete Mathematics”, Schaum‘s Outlines, Tata McGraw Hill Pub. Co. Ltd., New Delhi, 2021.
3. T. Koshy, “Discrete Mathematics with Applications”, Elsevier Publications, 2015.
4. S B Singh. “Discrete Structures”, Khanna Book Publishers Co-Pvt. Ltd. 2019.

SKILLS:

- ✓ *Identifying identity elements of an Algebraic structure and inverses of elements.*
- ✓ *Evaluate the rank and nullity of a Linear Transformation.*


Prism Engineering Education Series
DISCRETE MATHEMATICAL STRUCTURES
Sixth Edition
For Third Semester B.E. Classes (CSE and ISE Branches)
As per Revised VTU Syllabus 2019-20
Dr. D.S.C
PRISM

ImageSource:https://www.amazon.in/Discrete-Mathematical-Structures-Dr-D-S-C/dp/9388478398

22MT107 DISCRETE MATHEMATICAL STRUCTURES

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Basics of set theory, Algebra.

COURSE DESCRIPTION AND OBJECTIVES:

The course objective is to provide students with an overview of discrete mathematics. Students will learn about of group and expected to demonstrate analytical and combinatorial methods such as propositional logic, Mathematical Induction, Boolean functions, combinatorial, recurrence relation, generating function and graph theory

MODULE-1

UNIT-1 **8L+8T+0P=16 Hours**

BOOLEAN ALGEBRA AND LOGIC:

Group: Group, Subgroup, Lagrange's theorem, Introduction to rings and fields.

Boolean algebra: Boolean algebra, Truth table, Basic logic gates, Postulates of Boolean algebra, Principle of duality, Propositions, Connectives, Equivalence and Normal form.

UNIT-2 **8L+8T+0P=16 Hours**

APPLICATIONS OF LOGIC & BOOLEAN ALGEBRA:

Normal form: CNF, DNF, CDNF, CCNF, Conversion of CNF to DNF and vice versa.

Minimization of Boolean function (Karnaugh Maps), Generalized Pigeonhole Principle.

PRACTICES:

- Determine order of an element of group.
- Determine the order of a subgroup of a finite group.
- Determine whether algebraic structure is a Ring or a Field.
- Construct Truth table of propositions.
- Check whether propositions are equivalence.
- Obtain CNF, DNF of expression.
- Draw 2 –variables, 3-variables K-map.
- Minimize the Boolean function by K-map.
- Simplify the Boolean expression using Boolean algebra laws.
- Determine the homogeneous solution and particular solution for recurrence relation.
- Switching Circuit in Boolean algebra, Combination of two switches in a Circuit

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

COMBINATORICS AND GRAPHS:

Combinatorics: The basics of counting, Permutations and combinations, Discrete Numeric Functions. Recurrence relations and Generating functions.

Graph theory: Graph terminology, Special types of graphs, Connected graph, Weighed graph, Graph Isomorphism, Euler and Hamiltonian paths and circuits, Planar graphs, Bipartite graph, Tree.

UNIT-2 **8L+8T+0P=16 Hours**

APPLICATIONS OF COMBINATORICS AND GRAPHS:

Graph Coloring, Chromatic number, Matrix representation of graph, DFS, BFS algorithms, Minimum spanning tree.

PRACTICES:

- Represent sequence by Generating function.
- Solve Generation function by recurrence relation.
- Find degree of digraph and undirected graph.
- Identify Isomorphic graph, Euler circuit, Hamiltonian circuit.
- Determine matrix representation of graph.
- Determine the regions of planer connected graph.
- Draw Kn, Km,n.
- Determine the chromatic number of graphs.
- Find the weight of minimum spanning tree.
- Find number of arrangements that are possible.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply fundamental concepts of logic, set theory, and mathematical reasoning to solve discrete problems.	PO1, PO2, PO9, PO10, PO12, PSO1
CO2	Analyze relations, functions, and counting techniques to understand their applications in computer science and engineering.	PO1, PO2, PO9, PO10, PO12, PSO1
CO3	Evaluate recurrence relations, graph theory, and algebraic structures to model real-world computational problems.	PO1, PO2, PO9, PO10, PO12, PSO1
CO4	Create innovative solutions using discrete mathematical techniques for algorithm design and optimization.	PO1, PO2, PO9, PO10, PO12, PSO1

TEXT BOOKS:

1. Tremblay, J.P. and Manohar. R, "Discrete Mathematical Structures with Applications to Computer Science", 30th Reprint, Tata McGraw Hill Pub. Co. Ltd, New Delhi, 2017.
2. Rosen, K.H., "Discrete Mathematics and its Applications", 7th Edition, Tata McGraw Hill Pub. Co. Ltd., New Delhi, 2018.

REFERENCE BOOKS:

1. R.P. Grimaldi, "Discrete and Combinatorial Mathematics: An Applied Introduction", 4th Edition, Pearson Education Asia, Delhi, 2017.
2. S. Lipschutz and Mark Lipson., "Discrete Mathematics", Schaum's Outlines, Tata McGraw Hill Pub. Co. Ltd., New Delhi, 2021.
3. T. Koshy, "Discrete Mathematics with Applications", Elsevier Publications, 2015.
4. S B Singh. "Discrete Structures", Khanna Book Publishers Co-Pvt. Ltd. 2019.

SKILLS:

- ✓ *Familiarity of concepts of statements, logic and truth tables.*
- ✓ *Analyze closed form of discrete numeric function.*
- ✓ *Know some basic properties of graphs, trees and related discrete structures.*

Source: https://www.geeksforgeeks.org/best-way-to-start-with-competitive-programming-geeksforgeeks-cp-live-course/

22TP104 BASIC CODING COMPETENCY

Hours Per Week :

L	T	P	C
-	1	3	2

PREREQUISITE KNOWLEDGE: Programming in C.

COURSE DESCRIPTION AND OBJECTIVES:

This course is aimed to impart knowledge on advanced concepts of C programming language and problem solving. At the end of this course, students will be able to design, implement, test and debug complex problems using features of C.

MODULE-1

UNIT-1 **0L+4T+12P=16 Hours**

NUMBER CRUNCHING :

PRACTICES:

Problems On Number Crunching

- Write a program to check if a given number is perfect or not.
- Write a program to check if a given number is deficient or not.
- Write a program to check if 2 given numbers are amicable or not.
- Write a program to check if 2 given numbers are betrothed or not.
- Write a program to check whether a given number is an Armstrong number or not.
- Write a program to print the series of prime numbers in the given range.
- Write a program to print all the perfect numbers in a given range.
- Write a program to generate all deficient numbers in a given range.
- Write a program to generate all the amicable numbers in a given range.
- Write a program to generate all the betrothed numbers in a given range.
- Write a program to find the largest prime factor of a given number.
- Write a program to check whether the given number is a palindrome or not.
- Write a program to calculate sum of the individual digits for the given number.
- Write a program to find the first number that has more than ‘n’ factors, excluding 1 and that number.
- Write a program to accept a number as input and print its factorial.
- Write a program to accept a number n, print first N Fibonacci numbers.
- Write a program to check if an input number is Armstrong number or not.
- Write a program that takes input a,b. Print a power b.
- Write a program that takes input a number n, check if it a perfect square or not.
- Print array in spiral format.
- Print sum of each row in a matrix.
- Print sum of each column in matrix.
- Print left->right and right->left diagonals in a matrix.
- Initially you are at (0,0) find the shortest path count to reach the (n, n) block in matrix.
- Remove all the elements present in row and column of unsafe elements. An element is called unsafe if it is equal to smallest or largest value. Count number of remaining elements.
- Write a program to check if the string contains all the letters of alphabet.

- Check if a string is matching password requirements.
- Check if String A contains String B (String searching).
- Check if a number is harshad number or not.
- Write a program to get 3 numbers as input. The first is the number num1 and second is the digit that needs to be replaced. The third is the digit that is to replace the 2nd digit. Print the number after performing this operation.
- Write a program to accept a number and swap its alternate digits. Print the number generated.
- Write a program to accept a number and choice as input. If the choice is 0 rearrange the number such that the odd digits are ordered first followed by the even digits. If the choice is 1 rearrange the number such that the even digits are ordered first followed by the odd digits. Print the rearranged number. The order of occurrence of the digits is to be preserved.
- Write a program to determine that whether the given quadrilateral is cyclic or not. You are given the sizes of angles of a simple quadrilateral (in degrees) A, B, C and D, in some order along its perimeter.

  Note: A quadrilateral is cyclic if and only if the sum of opposite angles is 180^0.
- Chef is a very lazy person. Whatever work is supposed to be finished in x units of time, he finishes it in m*x units of time. But there is always a limit to laziness, so he delays the work by at max d units of time. Given x,m,d, find the maximum time taken by Chef to complete the work.
- Suppose Chef is stuck on an island and currently he has x units of food supply and y units of water supply in total that he could collect from the island. He needs xr units of food supply and yr units of water supply per day at the minimal to have sufficient energy to build a boat from the woods and also to live for another day. Assuming it takes exactly D days to build the boat and reach the shore, tell whether Chef has the sufficient amount of supplies to be able to reach the shore by building the boat? Read five integers x,y,xr,yr,D from the user and display “YES” if Chef can reach the shore by building the boat and “NO” if not (without quotes).
- There are 3 problems in a contest namely A,B,C respectively. Alice bets Bob that problem C is the hardest while Bob says that problem B will be the hardest.

  You are given three integers SA,SB,SC which denotes the number of successful submissions of the problems A,B,C respectively. It is guaranteed that each problem has a different number of submissions. Determine who wins the bet.

  1) If Alice wins the bet (i.e. problem C is the hardest), then output Alice.

  2) If Bob wins the bet (i.e. problem B is the hardest), then output Bob.

  3) If no one wins the bet (i.e. problem A is the hardest), then output Draw.

  Note: The hardest problem is the problem with the least number of successful submissions.

  Input Format

  - The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows.
  - The first and only line of each test case contains three space-separated integers SA,SB,SC, denoting the number of successful submissions of problems A,B,C respectively.

  Output Format

  For each test case, output the winner of the bet or print Draw in case no one wins the bet.

  Sample Input 1

  ```
  3
  1 4 2
  16 8 10
  14 15 9
  ```

  Sample Output 1

  ```
  Draw
  Bob
  Alice
  ```

SKILLS:

- ✓ *Analysis of the problem to be solved.*
- ✓ *Application of various file operations effectively in solving real world problems.*
- ✓ *Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.*

- In a season, each player has three statistics: runs, wickets, and catches. Given the season stats of two players A and B, denoted by R, W, and C respectively, the person who is better than the other in the most statistics is regarded as the better overall player. Tell who is better amongst A and B. It is known that in each statistic, the players have different values.

  Input

  The first line contains an integer T, the number of test cases. Then the test cases follow.

  Each test case contains two lines of input.

  The first line contains three integers R1, W1, C1, the stats for player A.

  The second line contains three integers R2, W2, C2, the stats for player B.

  Output

  For each test case, output in a single line "A" (without quotes) if player A is better than player B and "B" (without quotes) otherwise.

- Write a program to find the direction.

  Chef is currently facing the north direction. Each second he rotates exactly 90 degrees in clockwise direction. Find the direction in which Chef is facing after exactly X seconds.

  Note: There are only 4 directions: North, East, South, West (in clockwise order). Initially chef is at 0th second i.e., facing North direction.

  Input Format

  - First line will contain T, number of testcases. Then the testcases follow.
  - Each testcase contains of a single integer X.

  Output Format

  For each testcase, output the direction in which Chef is facing after exactly X seconds.

  Sample Input 1

  3

  1

  3

  6

  Sample Output 1

  East

  West

  South

- Chef is playing in a T20 cricket match. In a match, Team A plays for 20 overs. In a single over, the team gets to play 6 times, and in each of these 6 tries, they can score a maximum of 6 runs. After Team A's 20 overs are finished, Team B similarly plays for 20 overs and tries to get a higher total score than the first team. The team with the higher total score at the end wins the match.

  Chef is in Team B. Team A has already played their 20 overs, and have gotten a score of R. Chef's Team B has started playing, and have already scored C runs in the first O overs. In the remaining 20−O overs, find whether it is possible for Chef's Team B to get a score high enough to win the game. That is, can their final score be strictly larger than R?

  Input: There is a single line of input, with three integers, R, O, C.

  Output: Output in a single line, the answer, which should be "YES" if it's possible for Chef's Team B to win the match and "NO" if not.

- Make Array Zeros using pointers

  You are given an array A of length N (size should be created using Dynamic memory allocation) and can perform the following operation on the array:

  Select a subarray from array A having the same value of elements and decrease the value of all the elements in that subarray by any positive integer x.

  Find the least possible number of operations required to make all the elements of array A equal to zero.

The first line contains an integer N denoting the number of elements in the array.

The next line contains space-separated integers denoting the elements of array A.

Print the least possible number of operations required to make all the elements of array A equal to zero.

Sample Test case

Input:

```
5
2 2 1 3 1
```

Output:

```
4
```

UNIT-2 **0L+4T+12P=16 Hours**

PATTERNS

PRACTICES:

Problems on Number Patterns

- Write a program to generate Floyd triangle. Sample input N= 4.

```
1
2 3
4 5 6
7 8 9 10
```

- Write a program to generate the following pattern. Sample input N=5.

```
13579
3579
579
79
9
```

- Write a program to generate the following pattern. Sample input N=4.

```
1111111
222222
33333
4444
333
22
1
```

- Write a program to generate the following pattern. Sample input N=5.

```
5432*
543*1
54*21
5*321
*4321
```

- Write a program to generate the following pattern. Sample input N=5.

```
12     21
123   321
1234 4321
123454321
```

- Write a program to generate the following pattern. Sample input N=5.

```
1
2*2
3*3*3
4*4*4*4
4*4*4*4
3*3*3
2*2
1
```

- Write a program to generate the following pattern. Sample input N=4.

```
1
212
32123
4321234
```

- Write a program to generate the following pattern. Sample input N=5.

```
*
* *
* * * *
* *
*
```

- Write a program to print Pascal triangle for the given number of rows. Sample input N=5.

```
                    1
               1         1
          1         2         1
     1         3         3         1
  1       4         6         4         1
```

- Write a program to generate the following pattern. Sample input N=4.

```
1234
2341
3421
4321
```

- Print Hollow Diamond pattern.
- Print pascals triangle.
- Print Floyds triangle.
- Print Butterfly Pattern.
- Print palindromic pattern.
- Print full inverted number triangle.
- Check if a number is prime or not (Efficient Approach).
- Find sum of all the digits of the number.
- Print transpose of given matrix.
- Rotate a two dimensional matrix by 90, 180, 270 degrees.

MODULE-2

UNIT-1 **0L+4T+12P=16 Hours**

ARRAYS:

PRACTICES:

Problems On Arrays

- Given an unsorted array of size N, and the array elements are in the range of 1 to N. There are no duplicates, and the array is not sorted. One of the integers is missing. Write a program to find the missing number.
- Given an array consisting of only 0s and 1s in random order rearrange the array such that all the 0s are to the left of the array and 1s to the right.
- Give an array consisting of odd and even numbers in random order, rearrange the array such that all the odd numbers are to the left of the array and even numbers are to the right of the array.
- Write a program to find all the unique elements in an array.
- Write a program to merge two arrays of the same size sorted in descending order.
- Write a program to count the frequency of each element in an array of integers.
- Write a program to find the second largest element in an array.
- Write a program to find the second smallest element in an array.
- Write a program to find that one element in array that occurs odd number of times, where every other element appears even number of times.
- Create a jagged array (adjacency list representation of a graph) with no of rows and no of columns in each row as specified by the user.

  Hint: Use Dynamic memory allocation (malloc() or calloc())

  Input:

  ```
  Enter no of rows: 3
  Enter no of columns Row in 1: 3
  Enter no of columns Row in 2: 5
  Enter no of columns Row in 3: 2
  Enter the elements row wise:
  8 6 5
  8 4 6 9 7
  9 2
  ```

  Output:

  ```
  8 6 5
  8 4 6 9 7
  9 2
  ```

- Write a program to find second largest number in the array.
- Write a program to find first repeating element in the array.
- Write a program to left rotate the array.
- Write a program to right rotate the array.
- Write a program to find the largest continuous sum.
- Write a program to print the sum of 2nd largest and 2nd smallest elements.
- Write a program to find the maximum product of two numbers multiplies in array (same index should not be used twice).
- Rearrange an array consisting of 1s and 0s such that they are alternatively arranged. Print minimum number of moves required.
- In a given array, find two numbers whose sum equal k.
- Find the difference between positive and negative elements in the array.
- Implement sorting algorithms (Insertion, selection, bubble).

UNIT-2 **0L+4T+12P=16 Hours**

STRINGS:

PRACTICES:

Problems on Strings:

- Write a program to reverse a given string word by word.

- Write a program to find the first occurrence of non-repeating character in the given string.
- Write a program to compress the string as provided in the example.
- Write a program to expand a string as provided in the example.
- Write a program to reverse those words of a string whose length is odd.
- Write a program to check if a given matrix is symmetric or not.
- Write a program to convert all the cases of letter (Lower case -> Upper Case, Upper Case-> Lower Case).
- Write a program to reverse all the words (Not the entire sentence but individual words).
- Find the longest palindrome in a given string.
- Check if two strings are anagrams or not.
- Find minimum number of changes to be done to make a string palindrome.
- Convert Excel sheet name to number (A-1, B-2, Z-26, AA-27).
- Find number of possible palindromes present in a string.
- Write a C program to read a string s, and determine the number of words in s.

  Example : s=oneTwoThree

  There are 3 words in the string: ‘one’, ‘Two’, ‘Three’.
- Write a C program that reads a string S and remove all duplicates characters from the given string S.

  NOTE: 1) Order of characters in output string should be same as given in input string.

  2) String S contains only lowercase characters [‘a’-’z’].

  Example: S = Vignanuniversity

  The program should generate the output as: Vignauersty
- Today Ron is reading the book. Due to some reason, he started hating the word ‘are’ (without quotes). So he decided to replace the substring ‘are’ with ‘R’. Write a C program that reads a line of message ‘s’ and replace the substring ‘are’ with ‘R’. Example: s= Howareyou.

  The program should generate the output as: HowRyou
- Write a program to concatenate the characters of the two given strings alternatively.
- Given a string S consisting of uppercase and lowercase letters, change the case of each alphabet in this string. That is, all the uppercase letters should be converted to lowercase and all the lowercase letters should be converted to uppercase.

  Input: Vignan University

  Output: vIGNAN uNIVERSITY
- Write a program to insert a given character at the beginning and end of the given string.
- Given two Strings A and B. They are said to be friends if ASCII sum of the each individual string is divisible by 4 else they are not friends. You need to find whether given two strings are friends or not.

  Sample Test case:

  Input:

  man nam

  vignan university

  Output:

  YES

  NO
- Write a program to find the frequency of each digit in the given string.

  Input Format

  The first line contains a string, which is the given number.

  Output Format

  Print ten space-separated integers in a single line denoting the frequency of each digit, indicate that the integers are from 0 to 9.

Sample Input 0

a11472o5t6

Sample Output 0

0 2 1 0 1 1 1 1 0 0

Explanation 0

In the given string:

· 1 occurs two times.
· 2,4,5,6 and 7 occur one time each.
· The remaining digits and don't occur at all.

- Sherlock considers a string to be valid if all characters in the given string appear the same number of times. It is also valid if he can remove just 1 character at 1 index in the string, and the remaining characters will occur the same number of times.

  Write a C program that reads a string s and determine whether it is valid or not. If valid, return YES, otherwise return NO.

  Example: S=abc
  This is a valid string because frequencies are {a:1,b:1,c:1}
  S=abcc
  This is a valid string because we can remove one c and have 1 of each character in the remaining string.
  S=abccc
  This string is not valid as we can only remove 1 occurrence of c. That leaves character frequencies of {a:1,b:1,c:2}

- Read a string containing characters A and B only. Your task is to change it into a string such that there are no matching adjacent characters. To do this, you are allowed to delete zero or more characters in the string.

  Write a C program that finds the minimum number of deletions required.

  Example: S=AABAAB

  Remove A at positions 0 and 3 to make S=ABABA in 2 deletions.

  Input Format

  The first line contains an integer (the number of queries).

  The next q lines each contain a string s to analyze.

  Sample Input:

  ```
  5
  AAAA
  BBBBB
  ABABABAB
  BABABA
  AAABBB
  ```

  Sample Output:

  ```
  3
  4
  0
  0
  4
  ```

- Write a C program that reads a string 's' and it is said to be complete if it contains all the characters from a to z.
  Input Format
  First line of the input contains the number of strings N. It is followed by N lines each contains a single string.
  Output Format
  For each test case print "YES" if the string is complete, else print "NO"

Constraints 1 <= N <= 10
The length of the string is at max 100 & the string contains only the characters a to z.

- Write a C program that reads two strings and determine whether they share a common substring or not. A substring may be as small as one character.
  Example;
  S1=and
  S2=art
  The common substring in these two strings: a.
  Sample Input
  2
  hello
  world
  hi
  world
  Sample Output
  YES
  NO

COURSE OUTCOMES:

Upon successful completion of the course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply fundamental programming concepts to solve computational problems efficiently.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1
CO2	Analyze different coding techniques and debugging strategies to optimize program performance.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1
CO3	Evaluate various programming paradigms and their effectiveness in real-world applications.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1
CO4	Create structured and modular programs using advanced coding practices for problem-solving.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1

TEXT BOOKS:

1. Behrouz A. Forouzan, Richard F.Gilberg, "Programming for Problem Solving", 1st edition, Cengage publications, 2019.
2. Ajay Mittal, "Programming in C - A Practical Approach", 1st edition, Pearson Education, India, 2010.

REFERENCE BOOKS:

1. Reema Thareja, "Computer Fundamentals and Programming in C", 1st edition, Oxford University Press, India, 2013.
2. Herbert Schildt, "C: The Complete Reference", 4th edition, Tata McGraw-Hill, 2017.
3. Byron S Gottfried, "Programming with C", 4th edition, Tata McGraw-Hill, 2018.

22ME101 ENGINEERING GRAPHICS

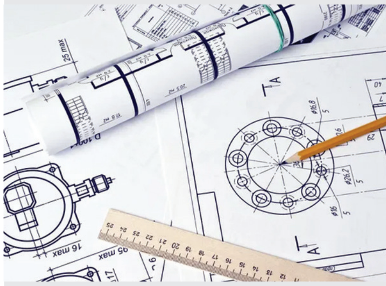


Image source: https://depositphotos.com/5087383/stock-photo-the-engineering-drawing.html

Hours Per Week :

L	T	P	C
2	-	2	3

PREREQUISITE KNOWLEDGE: Basics of Geometry

COURSE DESCRIPTION AND OBJECTIVES:

Engineering graphics is the language of engineers and is the most effective way of communicating and sharing technical ideas in the form of pictures/drawings. The objective of this course is to familiarize the students with the conventional concepts of engineering drawing and computer aided drawing.

MODULE-1

UNIT-1 **6L+0T+6P=12 Hours**

ENGINEERING CURVES:

Types of lines; Lettering, Dimensioning, Geometric constructions - lines, polygons (Angle, ARC, General and Inscribe in circle method), Conical curves (General method), Ellipse by Oblong method.

UNIT-2 **10L+0T+10P=20 Hours**

ORTHOGRAPHIC PROJECTIONS OF POINTS, LINES & PLANES:

Principles of projection; Projections of points; Projection of straight lines - Inclined to one plane, inclined to both planes; Projection of planes - Inclined to one plane.

PRACTICES:

- Construction of polygons using different methods (i.e. ARC, Angle, General).
- Inscribe a regular hexagon & pentagon in a circle of the given diameter.
- Tracing of conical curves (Ellipse, Parabola, Hyperbola) by using General Method.
- Draw the projections of the points situated in all the 4 quadrants.
- Draw the projections of a line when it is inclined to one plane (HP or VP).
- Draw the projections of a line when it is inclined to both the planes (HP &VP).
- Draw the projections of a plane when it is inclined to one plane (HP or VP).

MODULE-2

UNIT-1 **6L+0T+6P=12 Hours**

PROJECTIONS OF SOLIDS AND DEVELOPMENT OF SURFACES:

Projections of Solids: Projection of solids axis inclined to one reference plane - Prisms, pyramids, Cylinder and cone.

Development of Surfaces: Development of lateral surfaces of simple solids - Prisms, Pyramids, Cylinder and cone.

UNIT-2 **10L+0T+10P=20 Hours**

ORTHOGRAPHIC VIEWS AND DRAFTING USING COMPUTER PACKAGE:

Orthographic Views: Conversion of pictorial views into orthographic views.

Drafting Using Computer Package: Introduction to 2D modelling software - AutoCAD; Conversion of Isometric view into Orthographic views of simple castings; Conversion of Orthographic views into Isometric view of simple solids - Prisms, Pyramids, Cylinders and cones.

SKILLS:

- ✓ *Convert isometric views of objects into orthographic views and vice versa.*
- ✓ *Visualize the shape of the 3D components.*
- ✓ *Create pictorial views by using AutoCAD.*
- ✓ *Understand projections by visualization.*

PRACTICES:

- Draw the projections of Prisms, when they are inclined to one reference plane (HP or VP).
- Draw the projections of Pyramids, when they are inclined to one reference plane (HP or VP).
- Draw the projections of cylinder & cone, when they are inclined to one reference plane (HP or VP).
- Draw the complete surface development of prisms & pyramids with the given dimensions.
- Draw the complete surface development of cylinder & cone with the given dimensions.
- Draw the orthographic view's (i. e. front view, top view, and side view) of the given pictorial view of the sketches by using AutoCAD.
- Draw the Isometric view of simple solids (Prisms & Pyramids) by using AutoCAD.
- Draw the Isometric view of simple solids (Cylinder & Cone) by using AutoCAD.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply geometric construction techniques to create engineering curves and lettering using standard conventions.	PO1, PO2, PO5, PO12, PSO1
CO2	Analyze and construct projections of points, lines, planes, and solids using orthographic methods.	PO1, PO2, PO3,PO4, PO5, PO12, PSO1
CO3	Evaluate the accuracy and completeness of multi-view drawings and develop lateral surfaces for engineering solids.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1
CO4	Create 2D models and convert between orthographic and isometric projections using CAD software.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1

TEXT BOOKS:

1. J Hole, “Engineering Drawing”, 2nd edition, Tata McGraw-Hill, 2019.
2. N D Bhatt, “Engineering Drawing”, 53rd edition, Charotar Publication, 2014.

REFERENCE BOOKS:

1. Basant Agrawal and C.M. Agrawal “Engineering Drawing”, 2nd edition, Tata Mc Graw- Hill, 2018.
2. K L Narayana, “Engineering drawing”, 3rd edition, SciTech Publications, 2011.
3. Colin H. Simmons, Dennis E. Maguire, Manual of Engineering Drawing, 2nd edition, 2003.

22EN104 TECHNICAL ENGLISH COMMUNICATION

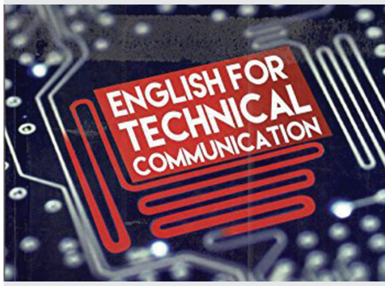


Image source: https://www.abebooks.com/9781316640081/English-Technical-Communication-Students-Book-1316640086/plp

Hours Per Week :

L	T	P	C
1	2	2	3

PREREQUISITE KNOWLEDGE: Basic sentence formation, understanding contextual meanings, basic writing skills and moderate fluency in English.

COURSE DESCRIPTION AND OBJECTIVES:

In this course students will read, analyze, and interpret material from technical and general fields, and practice reading, writing, listening and speaking skills to gain exposure and functional English on a variety of contemporary topics. The overall course objective is to provide English for Specific Purposes(ESP) instruction to enhance students' reading, writing, listening and speaking skills through a practice in the language. It will aim to build students' confidence and motivation through exposure to academic skills like Note making/taking, Paraphrasing, Summarizing, Report Writing, Making Presentations etc., so as to generate interest in the language from an ESP perspective. Finally, students are expected through the course to gain key strategies and expression for communicating with professionals and non-specialists.

MODULE-1

UNIT-1 4L+8T+8P=20 Hours

GENETICS:

Reading: Reading for Note Making Sub skills: Reading for global understanding (skimming), specific information (scanning), understanding main ideas and supporting ideas, guessing contextual meanings from the text. -Vocabulary building: commonly used roots, prefixes, and suffixes.

Writing: Note making, organising main points and sub points, numbering and sequencing, suggesting titles, paraphrasing and summarising.

Functional grammar: Common Errors in Articles and Prepositions (Handout).

Listening: Listening for Note Taking: top down and bottom up approach, listening for main ideas and supporting points.

Speaking: Presentation in teams - ideas on the topic summarised, making a PPT, effective introductions and conclusions, logical organisation of content, using appropriate structure and cohesive devices.

UNIT-2 4L+8T+8P=20 Hours

ALIENS:

Reading : Predicting, skimming, scanning, reading for inference, extrapolative reading

Vocabulary building: Academic vocabulary from the text: synonyms, antonyms, Words often confused.

Writing : Paragraph writing; writing a topic sentence, supporting sentences, effective introductions and conclusions, use of cohesive devices. Types of Paragraphs: Descriptive, narrative, argumentative and expository.

Functional grammar: Common Errors in Verb forms and Conditional sentences (Handout).

Listening : Listening for identifying parts from a description, listening to and sorting information, listening for specific information.

Speaking : Narrating/Retelling an incident, using suitable cohesive devices/discourse markers Speaking of past and present habits/ activities/events - Speaking of future plans.

PRACTICES:

- Note making.
- Summarizing.
- Paragraph Writing.
- Error correction and Restructuring.
- Vocabulary building.
- Listening comprehension.
- Note taking.

MODULE-2

UNIT - 1 **4L+8T+8P=20 Hours**

SOCIAL MEDIA – HEALTH AND NUTRITION:

Reading : Reading for factual information researching for supporting evidence - skimming, scanning, Vocabulary building: One-word substitutes.

Writing : Letter Writing- E-mail writing – New age communication – Format, protocol, and style- WhatsApp, Facebook and Twitter Functional grammar: Common Errors in Sub-Verb Agreement and Modals.

Listening : Listening to a Business Presentation: Listening for deducing information, for abstract details and specific details, listening for taking a message.

Speaking : Making a presentation with a PPT on a topic assigned- organising the presentation using appropriate discourse markers - presenting a point of view - Extempore.

UNIT-2 **4L+8T+8P=20 Hours**

FASHION:

Reading : Reading for data interpretation and information transfer from graphical aids to text reports (pictograms. tables, graphs, pie charts, flow charts), deducing specific information and general information

Vocabulary building: Business vocabulary, collocations, idioms and phrasal verbs.

Writing: Writing a Report: Drafting general and factual reports - writing an overview - an effective introduction - organising information into paragraphs (Stages of writing: planning /organising /writing / editing /rewriting)

Functional grammar: Transformations and miscellaneous common errors.

Listening : Listening to a Ted talk and sorting information – taking notes from a discussion.

Speaking : Group Discussion – prerequisites -generating content - initiating a discussion - expressing one's opinion ~ leading a discussion - agreeing/ disagreeing to someone's view - cutting into a speech - body language and voice modulation.

PRACTICES:

- E-mail writing.
- Letter writing.
- Report writing.
- Messaging in Social media.
- Extempore.
- Making PPTs.

SKILLS:

- ✓ *Apply different sub skills like skimming, scanning, reading for information, reading for inference etc. to understand different kinds of text.*
- ✓ *Apply different sub skills like top down, bottom up approaches to listening.*
- ✓ *Use functional vocabulary relevant to engineering and technology to express ideas lucidly.*
- ✓ *Use appropriate sentence structure, cohesive devices to construct simple text in regular correspondence like e-mails and letters.*

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply academic and functional language to speak fluently and present clearly on technical and professional topics.	PO1,PO2, PO9,PO10, PO12, PSO3
CO2	Apply appropriate grammar and structure to write coherent content for workplace communication and technical writing.	PO1,PO2, PO9,PO10, PO12, PSO3
CO3	Analyze and interpret a wide range of texts and audio materials related to technology and workplace communication.	PO1,PO2, PO9,PO10, PO12, PSO3
CO4	Evaluate listening strategies to comprehend technical content by identifying main ideas, tone, and contextual meaning.	PO1,PO2, PO9,PO10, PO12, PSO3

TEXT BOOK:

1. N P Sudharshana & C Savitha, “English For Technical Communication”, Cambridge University Press, 2016.

REFERENCE BOOKS:

1. Balasubramanian T,“A Text book of Phonetics for Indian Students”, Orient Longman, New Delhi, 1989.
2. Krishnaswamy, N and Sriraman, T, “Current English for Colleges”, Trinity publications, 2016.
3. Mohan Krishna and Meera Banerjee, “Developing Communication Skills”, Macmillan India Ltd. New Delhi, 1990.
4. Ashraf Rizvi M, “Effective Technical Communication”, 2nd Edition, McGraw Hill Education, 2017.
5. Narayana Swamy V R, “Strengthen your Writing”, Third Edition Orient Black Swan, New Delhi, 2005.

Image Source: https://honestproscons.com/wp-content/uploads/2021/03/web-technology.jpg

22IT101 WEB TECHNOLOGIES

Hours Per Week :

L	T	P	C
1	2	2	3

PREREQUISITE KNOWLEDGE: Computer Programming

COURSE DESCRIPTION AND OBJECTIVES:

This course offers the basic concepts used to develop static web pages and it also provides knowledge of Internet programming concepts. Further, this course is to build web applications using HTML, CSS, and client side script technologies that span multiple domains.

MODULE-1

UNIT-1 **4L+8T+8P=20Hours**

HTML BASICS:

Coding Basics: HTML Syntax, html, head, title, and body tags, Headings, paragraphs and lists, The strong and em tags, The doctype, The lang attribute, The meta tag, and the Unicode character set.

Coding Links: Absolute & Relative URLs: Anchor tags and hrefs, Linking to other websites, Linking to pages within a website, Opening a link in a new browser window/tab.

Adding Images: The break tag, The image tag, and source attribute, Using the width, height, and alt attributes, Using horizontal rules, tables, forms, and frames.

UNIT-2 **4L+8T+8P=20 Hours**

CASCADING STYLE SHEETS AND DIV TAGS:

Cascading Style Sheets (CSS): The style tag, Tag selectors, font size, font family, color, & line-height properties, and Hexadecimal color codes.

CSS Properties: Text, background, border, list and font.

CSS Class Selectors: The class attribute, CSS class selectors, The span tag, CSS opacity.

Div Tags, ID Selectors, & Basic Page Formatting: Dividing up content with the div tag, Assigning IDs to divs, Setting width & max-width, CSS background color, Adding padding inside a div, Centering content, CSS borders, CSS shorthand & the DRY principle.

PRACTICES:

- Practice Basic HTML tags
- Create links on same page and other pages
- Insert images on a web page
- Create lists on a web page
- Create Tables on a web page
- Create forms such as login form and registration form etc.
- Working with Frames
- Add different types of CSS to web pages
- Usage of div tag in the web page
- Create a personal website using HTML and CSS.

MODULE-2

UNIT-1 **4L+8T+8P=20 Hours**

JAVA SCRIPT:

Java Script: Introduction, Document Object Model, Language Syntax, Variable declaration, Operators, Control Statements, Understanding Arrays, Function Declaration.

Built-in Functions: Standard Date and Time, String, Array and Math.

UNIT-2 **4L+8T+8P=20 Hours**

HTML FORM VALIDATIONS:

HTML Form Validations: HTML Document Object Model, Working with HTML form and its elements.

Working with Objects and Classes: Working with Objects, Call method in JavaScript, Inheritance in JavaScript using prototype.

Java script Events: Keyboard events, mouse events, form events.

PRACTICES:

- Practice basic JavaScript programs such as the variable declaration and operators.
- Usage of Control Statements in JavaScript.
- Creating and accessing arrays in JavaScript.
- Working with functions in JavaScript.
- Perform validations on HTML forms using JavaScript.
- Working with Cookies.
- Create JavaScript Objects and Classes.
- Apply JavaScript on HTML and CSS webpages.

SKILLS:

✓ *Apply the CSS and JavaScript on HTML web pages.*

✓ *Develop static web pages.*

✓ *Create dynamic websites.*

COURSE OUTCOMES:

Upon successful completion of this course, students will have to ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply HTML syntax, tags, attributes, links, images, and tables to design basic web pages with structured content.	PO1, PO2, PO3, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO2	Analyze the use of CSS properties, class and ID selectors, and div tags to control page layout, styling, and responsiveness.	PO1, PO2, PO3, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO3	Evaluate the logic and flow of JavaScript programs using variables, arrays, functions, events, and built-in methods for web interactivity.	PO1, PO2, PO3, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO4	Create dynamic and user-friendly web pages with form validations, DOM manipulation, and object-based JavaScript event handling.	PO1, PO2, PO3, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3

TEXT BOOKS:

1. Jon Duckett, “Beginning Web Programming with HTML, XHTML, and CSS”, 2nd Edition, Wiley India Pvt. Ltd, 2010.
2. Julie C. Meloni , “HTML, CSS, and JavaScript All in One”, Sams Teach Yourself, 3rd Edition, Pearson, 2015.

REFERENCE BOOKS:

1. Chris Bates, “Web Programming, Building Internet Applications”, 3rd Edition, Wiley Dream Tech, 2012.
2. Jon Duckett, “HTML & CSS: Design and Build Websites”, 1st Edition, John Wiley & Sons, 2011.

II YEAR

B.Tech.

INFORMATION TECHNOLOGY

I SEMESTER

22ST202	-	Probability and Statistics
22CT201	-	Environmental Studies
22TP201	-	Data Structures
22MS201	-	Management Science
22IT201	-	Database Systems
22IT202	-	Digital Logic Design
22IT203	-	Object Oriented Programming
22SA201	-	Life Skills - I

II SEMESTER

22TP203	-	Advanced Coding Competency
22TP204	-	Professional Communication
22IT204	-	Design and Analysis of Algorithms
22IT205	-	Operating Systems
22IT206	-	Python Programming
	-	Department Elective-I
	-	Open Elective – 1
22SA202	-	Life Skills - II
	-	Minor / Honours - 1

COURSE CONTENTS

I SEM & II SEM

22ST202 PROBABILITY AND STATISTICS

Image Source: https://images.app.goo.gl/QBM6C8TQNTbNWXuA8

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Basic knowledge in statistics and mathematics.

COURSE DESCRIPTION AND OBJECTIVES:

To provide students with foundation in elementary topics of statistics and probability such as descriptive statistics, correlation, probability, random variables, correlation, regression, and testing of hypothesis. The course emphasizes statistics to solve engineering and management problems.

MODULE-1

UNIT-1 **12L+8T+0P = 20 Hours**

DESCRIPTIVE STATISTICS:

Basic Definitions, Frequencies, Graphical Representation, Histogram, Ogive curves; Measures of Central tendency, Arithmetic mean, Median, Mode, Mean deviation, Standard deviation; Symmetry and Skewness, Karl Pearson's Coefficient of skewness.

UNIT-2 **12L+8T+0P = 20 Hours**

PROBABILITY AND RANDOM VARIABLES:

Probability: Introduction, Definition (Classical and Axiomatic approach), Addition theorem, Conditional probability, Multiplication theorem and Bayes theorem.

Random Variables: Random variables, Discrete and Continuous variables and distribution function. Expectation, Variance of random Variables, Tchebysher's inequality.

PRACTICES:

- Various graphical presentation techniques.
- Measures of central tendency.
- Skewness.
- Karl Pearson's coefficient of skewness.
- Definitions of probability.
- Applications of addition theorem.
- Applications of multiplication theorem.

MODULE-2

UNIT-1 **12L+8T+0P = 16 Hours**

REGRESSION ANALYSIS AND DISTRIBUTIIONS:

Correlation and regression: Correlation, Types, Pearson's and Spearman's Coefficient of correlation, Regression, Regression lines.

Distributions: Introduction to Distributions: Binomial, Poisson and Normal distributions with properties and applications.

UNIT-2 **12L+8T+0P = 20 Hours**

TESTING OF HYPOTHESIS:

Testing large samples - single mean, two means, one proportion and two proportions. Testing small samples – single mean, two means (independent and paired samples), Chi square test -goodness of fit and independence of attributes.

PRACTICES:

- Correlation.
- Karl Pearson's coefficient of correlation.
- Regression and regression lines.
- Applications of statistical distributions.
- Testing the large sample tests-one mean and two sample means.
- One proportion and two proportion tests.
- Testing small samples-one, two samples and paired tests.
- Chi-square test for goodness of fit.
- Chi-square test for independence of attributes.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply descriptive statistics techniques to summarize and visualize data using central tendency, dispersion, and skewness measures.	PO1, PO2, PO3, PO4, PO6, PO12,PSO1
CO2	Analyze relationships among variables using correlation, regression, and curve fitting techniques.	PO1, PO2, PO3, PO4, PO6, PO12,PSO1
CO3	Evaluate probability-based models and discrete/continuous probability distributions for real-world decision-making scenarios.	PO1, PO2, PO3, PO4, PO6, PO12,PSO1
CO4	Create statistically valid conclusions through hypothesis testing using t-test, chi-square test, and large sample tests.	PO1, PO2, PO3, PO4, PO6, PO12,PSO1

TEXT BOOKS:

1. Sheldon M. Ross, An Introduction to Probability and Statistics for Engineers and Scientists, 3rd Edition, Academic Press, Elsevier.
2. S. C. Gupta and V. K. Kapoor, "Fundamentals of Mathematical Statistics", Sultan Chand & Sons, 2012.

REFERENCE BOOKS:

1. P. R. Vittal, "Mathematical Statistics", Margham Publications, Chennai, 2018.
2. Kishore S. Trivedi, "Probability and Statistics with Realiability, Queueing and Computer Science Applications", 2nd edition, Wiley Student edition, 2008.
3. A. Singaravelu, "Probability and Statistics", 22nd edition, Meenakshi Agency, 2015.

22CT201 ENVIRONMENTAL STUDIES

Image source: Biogas plant at VFSTR

Hours Per Week :

L	T	P	C
1	1	-	1

PREREQUISITE KNOWLEDGE: General awareness regarding environmental problems and importance of environmental protection.

COURSE DESCRIPTION AND OBJECTIVES:

It is a multidisciplinary subject where we deal with different aspects using a holistic approach. It is evolving to be the education for sustainable and ethical development both at a local and global level. It helps to prepare the next generation and to plan appropriate strategies for addressing environmental issues. It identifies and creates solutions that conserve to manage ecosystem and biodiversity and also helps to eliminate pollutants, toxicants to preserve air, water and soil quality. Environmental education recognizes impacts of global issues, enhances the public awareness and helps to take decisions towards environmentally responsible actions.

MODULE-1

UNIT-1 **4L+4T+0P=8 Hours**

INTRODUCTION TO ENVIRONMENT: NATURAL RESOURCES, ECOSYSTEMS AND BIODIVERSITY:

Environment and sustainable development; Natural resources- forest, water, energy and land resources; Ecosystem – basic structural components, function and interactions in ecosystem, ecological succession.

UNIT-2 **4L+4T+0P=8 Hours**

BIODIVERSITY AND CONSERVATION:

Introduction to biodiversity, types of biodiversity- species, genetic and ecosystem diversity; Threats to biodiversity - natural and anthropogenic, species extinctions, man wildlife conflicts; Biodiversity conservation - principles and strategies; in-situ and ex-situ conservation.

PRACTICES:

- Visit to a Biogas plant, Solar Power plant.
- Visit to a local area to document environmental assets – river / pond / lake / forest / grassland / hill / mountain.
- Set up an aquarium.
- Case study: Renewable energy use.

MODULE-2

UNIT-1 **4L+4T+0P=8 Hours**

ENVIRONMENTAL POLLUTION AND CLIMATE CHANGE:

Air, water, soil, radioactive and noise pollution; Study of different pollutants (SOx, NOx, PAN, PAH etc.); Toxicity study; Climate change - greenhouse effect, acid rain, ozone layer depletion.

UNIT-2 **4L+4T+0P=8 Hours**

POLLUTION CONTROL DEVICES AND WASTE WATER TREATMENT TECHNOLOGIES:

Air pollution control devices - Gravitational settling chambers, cyclonic separators, electrostatic precipitators, fabric filters and bio filters, Wastewater management.

SKILLS:

- ✓ *Create a biodiversity map of any habitat/ ecosystem.*
- ✓ *Strategize different ways of using renewable energy resources.*
- ✓ *Design novel strategies and approaches for pollution control and waste management.*

PRACTICES:

- Visit to a sewage treatment plant and waste-water analysis.
- Case study: Recycling Technologies.
- Case study: Effects of contaminants on microorganisms.
- Report writing: 12 principles of green chemistry for environmental sustainability.
- Report writing: Environmental Impact Analysis, Local Disaster Management Plan.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Gain an understanding of biodiversity conservation and environmental protection.	PO7, PO9, PO10, PO12
CO2	Analyze the sources of pollutants and their effects on the atmosphere.	PO6, PO8, PO9, PO11, PO12
CO3	Identify evidence of global warming, ozone depletion, and acid rain.	PO7, PO10, PO12
CO4	Develop a basic understanding of prevention, mitigation, preparedness, response, and recovery strategies.	PO6, PO8, PO9, PO11, PO12

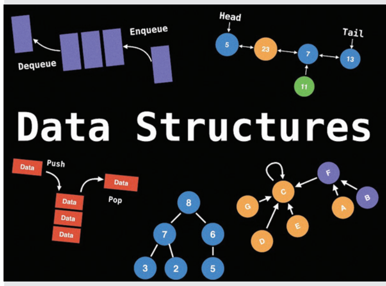
TEXT BOOK:

1. A. Kaushik and C. P. Kaushik, “Perspectives in Environmental Studies”, New Age International Publishers, 5th Edition, 2016.
2. Y. Anjaneyulu, “Introduction to Environmental Science”, B. S. Publications, 2015.

REFERENCE BOOKS:

1. B. Joseph, “Environmental Studies”, Mc Graw Hill Education, 2nd Edition, 2015.
2. S. Subash Chandra, “Environmental Science”, New Central Book Agency, 2011.
3. M. Basu and S. Xavier, “Fundamentals of Environmental Studies”, Cambridge University Press, 2016.
4. K. Mukkanti, “A Textbook of Environmental Studies”, S. Chand Company Ltd., 2009.
5. M. Anji Reddy, “A Textbook of Environmental Science and Technology”, B. S. Publications, 2008.

22TP201 DATA STRUCTURES



Source: https://www.youtube.com/watch?v=Qmt0QwzEmh0

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Programming in C.

COURSE DESCRIPTION & OBJECTIVES:

This course is aimed at offering fundamentals concepts of data structures and explains how to implement them. It begins with the basic concepts of data, data structures and then introduces the primitive and non-primitive data structures in detail. It forms the basis for understanding various ways of representing data and its usage in different computing applications.

MODULE-1

UNIT-1 **5L+6T+6P = 17 Hours**

DATA STRUCTURES BASICS:

Basic Terminology – data, information, datatype; Data Structures – Introduction, storage structures-sequential and linked storage representations; classification of data structures; Applications of data structures.

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort.

Searching: Linear Search and Binary Search.

UNIT-2 **11L+10T+10P = 31 Hours**

LINKED LISTS AND STACKS, QUEUES:

Linked List: Introduction, Types of linked list – Singly linked list, doubly linked list and circular linked list, representation of linked list, Operations of linked list: Traverse forward/ reverse order, searching, insertion and deletion; Applications of linked lists.

Stack – Introduction, array and linked representations, implementation and their applications; Queue – Introduction, array and linked representations, implementation; Types – Linear, circular and doubly ended queues – operations; Applications of Queues.

PRACTICES:

Problems on Recursion – Level 1

- Find the product of 2 numbers using recursion.
- Find the sum of natural numbers using recursion.
- Find the factorial of a number using recursion.
- Find the Nth term of Fibonacci series using recursion.
- Calculate the power using recursion.
- Write a recursive program for checking if a given number is a prime number.
- Given two integers write a function to sum the numbers without using any arithmetic operators.
- Convert a decimal to binary using recursion.
- Print all factors using recursion.
- Find the maximum product of digits among numbers less than or equal to N.

SKILLS:

- ✓ *Experienced to Store data and various types of data to handle.*
- ✓ *Ordering and sorting of data.*
- ✓ *Indexing and Searching of required data from large data sequences.*
- ✓ *Exposed to various characteristics such as Linear or non-linear, Homogeneous or heterogeneous and Static and Dynamic.*

Problems Recursion – Level 2

- Implement insertion sort recursively.
- Write a program to find the numbers less than N that are product of exactly 2 distinct prime numbers - using recursion.
- Implement selection sort recursively.
- Find the middle of a singly linked list using recursion.
- Find the sum of even numbers of an array using recursion.
- Check if a given array is in sorted order using recursion.
- Print alternate nodes of a linked list using recursion.
- Reverse a doubly linked list using recursion.
- Write a recursive function that returns all permutations of a given list.
- Implement bubble sort recursively.

Problems on Sorting and Searching – Level 1

- Implement the insertion sort function.
- Implement the bubble sort function.
- Implement the quick sort function.
- Implement the merge sort function.
- Implement the selection sort function.
- Implement linear search function.
- Implement binary search function.

Problems on SLL – Level 1

- Implement the insert function to insert nodes into a singly linked list (ascending order).
- Implement the insert function to insert nodes into a singly linked list (descending order).
- Implement the search node function.
- Implement the delete node function.
- Display forwards function.
- Display backwards function.
- Count the number of nodes in a singly linked list.
- Swap alternate nodes of a singly linked list.
- Move last node to the front of the linked list.
- Move first node to the last of the linked list.

Problems on Stacks – Level 1

- Implement two stacks using a single array.
- Given an array replace every element with nearest greater element on the right.
- Given a stack reverse the elements using only push and pop functions.
- Postfix evaluation using stack.
- Balance symbols.
- Find middle element in a stack.
- Remove middle element from a stack.
- Implement push and pop using linked list.
- Given an array of characters with the middle marked by X, check if the string is a palindrome.
- Maximum sum in sliding window.

Problems on Queues – Level 1

- Write a program to accept two numbers as input check if they are equal.
- Write a program to accept two characters as input and check if they are equal.
- Write a program to accept two numbers as input and print the greater of the 2 numbers.
- Write a program to accept two numbers as input and print the lesser of the 2 numbers.
- Write a program to accept 3 numbers as input and print the maximum of the 3.
- Write a program to accept 3 numbers as input and print the minimum of the 3.
- Write a program to accept a number as input and print EVEN if it is an even number and ODD if it is an odd number.
- Write a program to accept a number as input and check if it is divisible by 3. If it is divisible by 3 print YES else print NO.
- Write a program to accept a number as input and check if it is divisible by both 3 & 5. If it is divisible print YES else print NO.
- Write a program to accept a number as input and check if it is positive, negative or zero.

Problems on DLL – Level 1

- Implement insert function.
- Implement display forward function.
- Implement display backward function.
- Implement search function.
- Implement delete function.
- Reverse a doubly linked list from M to N.
- Find the sum of the odd and even nodes.
- Count odd keys of the linked list.
- Merge two sorted lists.
- Delete adjacent duplicate nodes.

Problems on CLL – Level 1

- Insert function (circular doubly linked list).
- Search function.
- Display forward.
- Display backward.
- Delete node (circular doubly linked list).
- Print the middle N nodes of a circular singly linked list.
- Move the last node of a circular singly linked list to the beginning.
- Delete adjacent duplicate nodes of a circular singly linked list.
- Delete nodes greater than a value from a circular doubly linked list.
- Find the sum of the nodes of a circular linked list.

Problems on Linked List – Level 2

- Given 2 sorted linked lists, print the common elements.
- Reverse a list (using Stack).
- Given a pointer to a node (not the last node), delete the node.
- Reverse a list (Recursive).
- Reverse a list (Iterative).
- Reverse a singly linked list in pairs (recursive).
- Reverse a singly linked list in pairs (iterative).
- Check if a singly linked list is a palindrome or not.
- Remove the loop if exists.
- Given 2 linked lists with data in the ascending order, merge them into a single list.

MODULE-2

UNIT-1 **8L+8T+8P=24 Hours**

TREES:

Trees: Basic Terminology, Types of Trees, Binary Tree – Introduction, properties, array and linked representations; Tree traversals and their implementation; Expression trees; BST – definition and operations, AVL trees – definition and construction; Applications of binary trees.

UNIT-2 **8L+8T+8P=24 Hours**

GRAPHS & HASHING:

Graphs: Basic Terminology, Types of Graphs, Graphs representations – adjacency matric, adjacency list; Traversals - breath first search and depth first search; Applications of graphs.

Hashing: Introduction, Different hash functions, collision: avoidance and handling methods.

PRACTICES:

Problems on BST – Level 1

- Insert function.
- Insert function (recursive).
- Search function.
- Pre order traversal.
- Post order traversal.
- In order traversal.
- Level order traversal.
- Delete child node.
- Delete parent node.
- Delete nodes greater than a value from a circular doubly linked list.

Problems on Priority Queues – Level 1

- Meeting rooms problem.
- Ugly number.
- Find median from data stream.
- Find the top K frequent elements.
- Find K Pairs with smallest sums.
- Find the Kth smallest element in a sorted matrix.
- Trapping Rain Water.
- Rearrange String k distance apart.
- Sort characters by frequency.
- Solve the maze problem.

Problems on Graphs – Level 1

- Implement Graph data structure.
- Implement BFS - iterative solution.
- Implement BFS - recursive solution.
- Implement DFS - iterative solution.
- Implement DFS - recursive solution.
- Check if given graph is strongly connected or not.
- Check if given graph is strongly connected or not - using DFS.
- Given a graph find the arrival and departure time of its vertices in DFS. Arrival time is the time when the vertex was explored for the first time, and departure time is the time at which all the neighbours are explored and are ready to backtrack.
- Given a directed acyclic graph and a source vertex, find the cost of the shortest path from source vertex to all other vertices present in the graph. If a vertex cannot be reached from given source vertex that distance may be printed as infinite.
- Given an undirected graph, check if the graph is 2 edge connected or not.

Problems on Hashing – Level 1

- Print a binary tree in vertical order.
- Find whether an array is subset of another array.
- Given an array A [] and a number x, check for pair in A [] with sum as x.
- Minimum operation to make all elements equal in array.
- Maximum distance between two occurrences of same element in array.
- Check if a given array contains duplicate elements within k distance from each other.
- Find duplicates in a given array when elements are not limited to a range.
- Most frequent element in an array.
- Smallest subarray with all occurrences of a most frequent element.
- First element occurring k times in an array.

Problems on Graphs – Level 2

- Find the shortest graph distances between every pair vertex in a given path. Assume that the graph does not have any negative edges.
- Find the shortest graph distances between every pair of vertices in a given path. The graph can have negative edges.
- Detect cycle in DFS.
- Count the number of connected components of a graph represented in the adjacent matrix.
- Count the number of connected components of a graph represented in the adjacent matrix - using DFS.
- Find a spanning tree - not necessarily a minimum spanning tree.
- Detect cycle in an undirected graph.
- Given an undirected graph, find its depth.
- Determine if a directed graph has a unique topological ordering.
- Given a directed acyclic graph and two vertices v and w, find the lowest common ancestor.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply appropriate data structures such as arrays, linked lists, stacks, queues, trees, and graphs to solve computational problems.	PO1, PO2, PO3, PO4, PO6, PO8, PO9, PO10, PO12, PSO1, PSO2
CO2	Analyze the behavior and efficiency of sorting, searching, and traversal algorithms across various data structures.	PO1, PO2, PO3, PO4, PO6, PO8, PO9, PO10, PO12, PSO1, PSO2
CO3	Evaluate the suitability of different linear and non-linear data structures like stacks, queues, trees, and graphs for specific real-world applications.	PO1, PO2, PO3, PO4, PO6, PO8, PO9, PO10, PO12, PSO1, PSO2
CO4	Create optimized data structure-based solutions using advanced structures such as AVL trees, hashing, and graph algorithms for complex problems.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO12, PSO1, PSO2

TEXT BOOKS:

1. Reema Thareja, “Data Structures Using C”, 2nd Edition, Oxford University Press, 2014.
2. Seymour Lipschutz, “Data Structures with C”, 1st Edition, McGraw Hill Education, 2017.

REFERENCE BOOKS:

1. Ellis Horowitz and Sartaj Sahni, “Fundamentals of Data Structures”, illustrated edition, Computer Science Press, 2006.
2. Richard F. Gilberg and Behrouz A. Forouzan, “Data Structures: A Pseudocode Approach with C”, 2nd Edition, CENAGE Learning, 2005.
3. R G Dromey and Pearson, “How to solve it by Computer”, 2nd edition, Impression edition, 1998.

22MS201 MANAGEMENT SCIENCE

Image Source: https://xueqi326.wordpress.com/semester-3/management-science/

Hours Per Week :

L	T	P	C
2	2	-	3

PRE-REQUISITE KNOWLEDGE: Basic knowledge of management

COURSE DESCRIPTION AND OBJECTIVES:

The goal of this course is to analyze the importance of management, significance of operation management and carry out production operations through work-study. Students will be able to analyse the markets, customers, competitors, and then plan HR function effectively. These management practices, functional areas of the organisation will helps the students to build up their career in the corporate world.

MODULE- 1

UNIT- 1 **6L+6T+0P =12 Hours**

INTRODUCTION TO MANAGEMENT:

Concepts of Management and organization- nature, importance and Functions of Management, Systems approach to Management - Taylor's Scientific Management Theory, Fayol's Principles of Management, Mayo's Hawthorne Experiments, Maslow's Theory of Human Needs, Douglas McGregor's Theory X and Theory Y, Herzberg's Two-Factor Theory of Motivation, Leadership Styles, Social responsibilities of Management.

UNIT- 2 **10L+10T+0P =20 Hours**

OPERATIONS MANAGEMENT:

Principles and Types of Plant Layout-Methods of production (Job, batch and Mass Production), Work Study -Basic procedure involved in Method Study and Work Measurement, Objectives, Need for Inventory control, EOQ, ABC Analysis, Purchase Procedure, Stores Management and Stores Records. Statistical Quality Control: control charts for variables and attributes (simple problems).

PRACTICES:

- Collect some examples with videos for types of production.
- Carry out production operations through work-study.
- Practice problems with Inventory control methods and Quality Control charts.

MODULE- 2

UNIT- 1 **8L+8T+ 0P =16 Hours**

HUMAN RESOURCES MANAGEMENT:

Concepts of Human Resource Management, Basic functions of HR Manager: Manpower planning, Recruitment, Selection, Training and Development, Placement, Wage and Salary Administration, Promotion, Transfer, Separation, Performance Appraisal, Grievance Handling and Welfare Administration, Job Evaluation, and Merit Rating.

UNIT- 2 **8L+8T+0P =16 Hours**

MARKETING MANAGEMENT:

Evolution of Marketing, Functions of Marketing Selling Vs Marketing, 4 P's of Marketing – Product Mix - Product Life Cycle – Place Mix – Channels of Distribution – Price Mix – Pricing Methods – Promotion Mix – Tools of Promotions.

PRACTICES:

- Select any Designation in an organization and try to describe its job description and job specifications.
- How do you deal with grievances at your work.
- Analyze marketing mix in various situations.

COURSE OUTCOMES:

Upon successful completion of this course, students will have to ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply management principles and motivational theories to identify the roles, responsibilities, and leadership styles in an organizational setup.	PO1, PO6, PO8, PO9, PO10, PO12, PSO3
CO2	Analyze various operations management functions such as plant layout, production types, inventory control, and quality control methods to improve organizational productivity.	PO1, PO2, PO4, PO11, PO12, PSO3
CO3	Evaluate the core functions of human resource management including recruitment, performance appraisal, and grievance handling to ensure employee satisfaction and organizational growth.	PO6, PO8, PO9, PO10, PO12, PSO3
CO4	Create strategic marketing plans using product, price, place, and promotion mix to meet customer needs and market demands.	PO1, PO6, PO10, PO11, PO12, PSO3

TEXT BOOKS:

1. Rajan Saxena: Marketing Management, 4th Edition, TMH, 2013.
2. Dilip Kumar Battacharya, Principles of Management, Pearson, 2012.

REFERENCE BOOKS :

1. Philip Kotler, Kevin Lane Keller, Abraham Koshy and Mithleshwar Jha: Marketing Management, 13th Edition, Pearson Education, 2012.
2. Dipak Kumar Bhattacharyya, Production and Operations Management, Universities Press, 2012.
3. Gary Dessler, “Human Resource Management”, 12th Edition, Pearson- 2012.
4. K.Aswathappa, “Human Resource Management”, Text and Cases”, TMH, 2011.
5. Harold Koontz, Heinz Weihrich, A.R. Aryasri, Principles of Management, TMH, 2010.

SKILLS:

✓ *Expert in managerial skills.*

✓ *Maintain social relations.*

✓ *Evaluate pricing strategies.*

Source: https:// encryptedtbn0. gstatic. com/ images?q=tbn: ANd9GcRKRy21 fK53Uz0lX3pal 8qFLxhXey Hkuhww0Q&usqp=CAU

22IT201 DATABASE SYSTEMS

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Computer Programming and Data Structures

COURSE DESCRIPTION AND OBJECTIVES:

This course presents an introduction to database management systems with an emphasis on how to organize, maintain and retrieve data efficiently from a relational database. It also focuses on requirements gathering and conceptual, logical, physical database design. The objective of the course is to enable the student to understand database design, expressing queries using SQL, transaction processing.

MODULE-1

UNIT-1 **12L+0T+8P=20 Hours**

DATA BASE DESIGN:

Databases and Database Users: Introduction; Characteristics of the database approach; Actors on the scene; Advantages of using the DBMS approach.

Database System Concepts and Architecture: Data models, Schemas, and instances; Three-Schema architecture and data Independence; Database languages and interfaces; The database system environment; Centralized and Client-Server architectures for DBMS.

Conceptual Data Modeling and Database Design: Entity types, Entity sets, Attributes, and keys; Relationship types, Relationship Sets, Roles, and structural constraints; Weak entity types; Relationship types of degree higher than two.

The Relational Data Model and Relational Database Constraints: Relational model concepts; Relational model constraints and Relational database schemas.

Basic SQL: SQL data definition and data types; Specifying constraints in SQL, Basic retrieval queries in SQL.

UNIT-2 **12L+0T+8P=20 Hours**

MORE SQL:

Complex Queries, Triggers, Views: More complex SQL retrieval queries; Views (virtual tables) in SQL, Introduction to PL/SQL Procedures, Functions Specifying constraints as assertions and actions as triggers.

PRACTICES:

- Design Conceptual database schema using ER Modeling Software Tools.
- Development of Relational Database schemas for Company/Student/Sailors/ using SQL and Perform the simple CURD Operations and create Database users give the permissions for access the database.
- Development of Relational Database schemas by specifying different types of Constraints for Company/Student/Sailors/Library perform Aggregate Functions expressing Nested queries using SQL and apply clauses like GROUP BY, HAVING and ORDER BY clauses of SQL.
- Design and Development of Company/Student/Sailors database and specifying queries using different types of JOINs.
- Creation and dropping of VIEWS for Company/Student/Sailors Data Bases.
- Implementation of PL/SQL programs with Control Structures.

- Implementation of Procedure for computes the square of value of a passed value.
- Implementation of Function for finding total number of students in class.
- Implementation of Triggers for updating employee salary details.

MODULE-2

UNIT-1 **12L+0T+8P=20 Hours**

BASICS OF FUNCTIONAL DEPENDENCIES AND NORMALIZATION FOR RELATIONAL DATABASES:

Informal design guidelines for relation schemas; Functional dependencies-inference rules, equivalence and minimal cover; Normal forms based on primary keys; Boyce-Codd normal form; multivalued dependency and 4NF; Join dependencies and 5NF; Properties of relational decompositions.

UNIT-2 **12L+0T+8P=20 Hours**

INTRODUCTION TO TRANSACTION PROCESSING CONCEPTS AND THEORY:

Introduction to transaction processing; Transaction and system concepts; Desirable properties of transactions; Characterizing schedules based on serializabilit.

Concurrency Control Techniques: Two-phase locking techniques for concurrency control; Concurrency control based on timestamp ordering.

PRACTICES:

- Implement the conversion of the database to first; second and third normal form is appropriate in a
- College environment.
- Library Management.
- Car Insurance.
- Hostel Management.
- Hospital Management.
- Transactions & TPS Activities in Indian Railways

Passenger Reservation System

a) Checking current position on any train

b) Make a reservation from any origin to any destination in India

c) Pay for the ticket using any option like debit card, credit card, etc

d) Changing/ Cancellation of bookings and getting money refund

Freight Operation Information System

a) Checking real-time position of goods trains

b) Checking number of rakes being used and number of idle rakes

c) Volume and weight of goods being carried

Crew Management System

a) Monitoring of activities of crew members

b) Checking the location, time of arrival and waiting time of crew members

c) Maintaining records of payment schedule information, duty schedule, training schedule of the crew members

Integrated Coach Management System

a) Checking real-time location and movement of trains

b) Scheduling trains movement near bottlenecks and heavy traffic zones

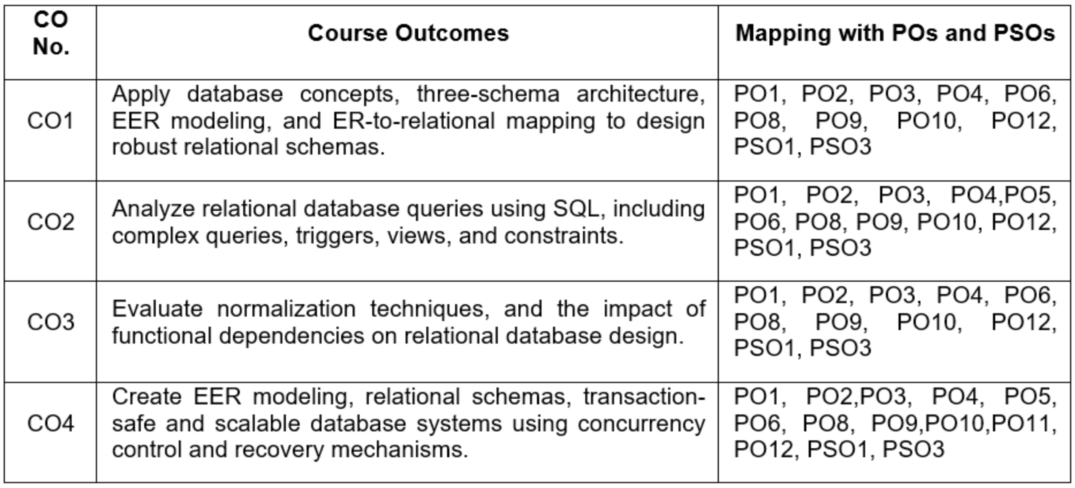
c) Improving train arrival time accuracy

SKILLS:

- ✓ *Conceptualize data using different data models.*
- ✓ *Translate data models into relational table schema (DDLs).*
- ✓ *Single block SQL queries and correlated subqueries.*
- ✓ *Functional dependencies and their relationship to keys.*
- ✓ *Find keys given a set of functional dependencies.*

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply database concepts, three-schema architecture, EER modeling, and ER-to-relational mapping to design robust relational schemas.	PO1, PO2, PO3, PO4, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3
CO2	Analyze relational database queries using SQL, including complex queries, triggers, views, and constraints.	PO1, PO2, PO3, PO4,PO5, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3
CO3	Evaluate normalization techniques, and the impact of functional dependencies on relational database design.	PO1, PO2, PO3, PO4, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3
CO4	Create EER modeling, relational schemas, transaction-safe and scalable database systems using concurrency control and recovery mechanisms.	PO1, PO2,PO3, PO4, PO5, PO6, PO8, PO9,PO10,PO11, PO12, PSO1, PSO3

TEXT BOOKS:

1. Ramez, Elmasri and Shamkant B. Navathe, “Fundamentals of Database Systems”, 7th Edition, Pearson Education, 2016.
2. Hector Garcia Molina, Jeffrey Ullman, Jennifer Widom, “Database Systems: The Complete Book”, 2nd Edition, Pearson Prentice Hall, 2009.

REFERENCE BOOKS:

1. Raghu Rama Krishnan and Johannes Gehrke, “Database Management Systems”, 3rd Edition, Tata McGraw Hill, 2013.
2. Abraham Silberschatz, Henry F.Korth and S.Sudarshan, “Database System Concepts”, 6th Edition, Tata McGraw Hill, 2010.
3. Corlos Coronel, Steven Morris, Peter Robb, “Database Principles Fundamentals of Design Implementation and Management”, 1st Edition, Cengage Learning, 2014.

22IT202 DIGITAL LOGIC DESIGN

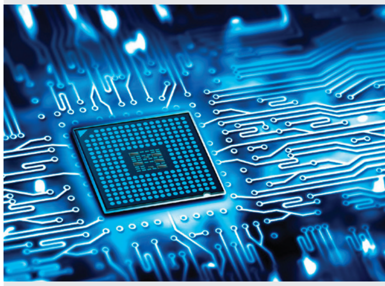


Image source: https://collegeacademy.in/images/ex-blogs/deld.jpg

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Basic Mathematics

COURSE DESCRIPTION AND OBJECTIVES:

The course addresses the concepts, principles and techniques of designing digital systems. The objective of this course is to offer the knowledge and skill of conversions between different number systems, design of logical gates, minimization of switching functions, design of combinational and sequential circuits, effective memory utilization and design of synchronous and asynchronous counters.

MODULE-1

UNIT-1 **6L+6T+0P=12 Hours**

NUMBER SYSTEMS:

Number systems, complements of numbers, Binary codes, Basic theorems and Properties of Booean Algebra, Boolean functions, Canonical and standard forms, Digial logic gates, The Map Method, Don't care conditions.

UNIT-2 **10L+10T+0P=20 Hours**

SIMPLIFICATION OF BOOLEAN FUNCTIONS:

Number base conversions, Simplification of Boolean expressions using Boolean algebra. optimal gate-level implementation of the Boolean functions using K-Map method, Gate implementation of Boolean functions using NAND and NOR.

PRACTICES:

- Number base conversions.
- Find r's complement and r-1's complements for the given numbers.
- Perform Subtraction using 2's complement.
- Convert gray to binary and binary to gray.
- Perform BCD subtraction.
- Simplify Boolean expressions using Boolean algebraic laws.
- Convert between canonical forms.
- Simplification of Boolean expressions using K-Map method.
- Solve problems on Don't care combinations.

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

COMBINATIONAL AND SEQUENTIAL CIRCUITS:

Combinational circuits, Analysis and design procedure, Binary adder– subtractor, Decimal adder, Binary multiplier, Magnitude comparator, Decoders, Encoders, Multiplexers, De-mutliplexers. Latches, Flip-flops, Analysis of clocked sequential circuits, Registers, Counters. RAM and types of RAM, ROM, Types of ROM, and PLDs.

SKILLS:

- ✓ *Students are able to convert numbers from one base to other.*
- ✓ *Students will be able to simplify the Boolean functions using Boolean algebraic laws and K-Map methods.*
- ✓ *Students are able to design combinational circuits and Sequential circuits.*
- ✓ *Students are able to realize Boolean functions using PROM, PAL and PLA.*

UNIT-2 **8L+8T+0P=16 Hours**

COUNTERS AND PLDS:

Design of code converters, realization of combinational circuits using decoders, multiplexers. Flip-flop conversions, Design of various types of synchronous and asynchronous counters, Implementation of Boolean functions using PAL and PLAs.

PRACTICES:

- Design the combinational circuits based on requirements using logic gates.
- Design the code converters (Binary to Gray…).
- Realize the Boolean expressions using NAND and NOR gates.
- Perform flip-flop conversions.
- Design the Asynchronous counters and Synchronous counters.
- Realize the Boolean functions using PROM, and PLA.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply number systems, binary codes, Boolean algebra, and logic gates to design and simplify digital circuits.	PO1, PO2, PO9, PO12, PSO1, PSO2
CO2	Analyze combinational circuits such as adders, subtractors, multiplexers, and decoders using standard design procedures.	PO1, PO2, PO3, PO4, PO9, PO10, PO12 PSO1, PSO2
CO3	Evaluate the use of memory devices and programmable logic components for implementing digital systems	PO1, PO2, PO3, PO4, PO10, PO12, PSO1, PSO2
CO4	Design sequential logic systems using flip-flops, counters, and registers based on specific timing and design requirements.	PO1, PO2, PO3, PO4, PO5,PO9, PO11, PO12, PSO1, PSO2

TEXT BOOKS:

1. M Morris Mano and Michael D. Ciletti, "Digital Design", 6th Edition, Pearson Education, 2018.

REFERENCE BOOKS:

1. H Taub and D Schilling, "Digital Integrated Electronics", 2nd Edition, TataMc Graw-Hill, 2004.
2. Z. Kohavi, "Switching and Finite Automata Theory", 2nd Edition, TataMc Graw-Hill, 2008.
3. C. V. S. Rao, "Switching and Logic Design", 3rd Edition, Pearson Education, 2009.

22IT203 OBJECT ORIENTED PROGRAMMING

Source of Image: https://www.datasciencecentral.com/what-should-java-developers-learn-in-2021/

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Computer Programming

COURSE DESCRIPTION AND OBJECTIVES:

This course is about the fundamentals of Object-Oriented Programming (OOP) Concept and OOP-based software development methodology. Java as a class-based and pure OOP language is used to demonstrate and implement appropriate concepts and techniques. The students are exposed to the concepts, fundamental syntax, and the thought processes behind object oriented programming. By end of the course, students will acquire the basic knowledge and skills necessary to implement object-oriented programming techniques in software development using Java.

MODULE-1

UNIT-1 **16L+0T+8P=24 Hours**

BUILDING BLOCKS OF OOPS:

Introduction: Java buzzwords, OOP principles, Data types, Operators, Control statements, Type conversion and casting, Arrays.

Classes and Methods: Introduction to classes and methods, objects, Constructors, Usage of static, Access control, String class, String Tokenizer.

Inheritance: Basics of Inheritance, Types of inheritance, Abstract classes, Interfaces, Usage of final, creating, defining and accessing Packages

UNIT-2 **8L+0T+8P=16 Hours**

OVERLOADING, INTERFACES AND PACKAGES:

Overloading: Overloading Methods and Constructors, this key word, Usage of super key word, Polymorphism, Method overriding

Interfaces: implementing interface, extending interfaces, accessing a package, importing packages.

PRACTICES:

- Reading different types of data from the user and display that data using Scanner class.
- Illustrating type conversions.
- Implementing different operators.
- Generating electricity bill
- Implementing different patterns.
- Implementing logical programs.
- Implementing Arrays.
- Implementing String class.
- Implementing String Tokenizer class.
- Implementing super keyword.
- Implementing forms of Inheritance
- Implementing overloading and overriding.
- Implementing runtime polymorphism.
- Create an abstract class Media (id, description). Derive classes Book (page count) and CD (play time). Define parameterized constructors. Create one object of Book and CD each and display the details.

SKILLS:

- ✓ *Analyze and develop algorithm for real life problems using Java.*
- ✓ *Able to develop multi-threaded applications.*
- ✓ *Able to create efficient software applications.*

- Define an interface, operations which has method area (), volume (). Define a constant PI having value 3.14. Create class a Cylinder which implements this interface (member-id, height). Create one object and calculate area and volume.
- Implementing packages.

MODULE-2

UNIT-1 **14L+0T+8P=22 Hours**

EXCEPTION HANDLING AND MULTITHREADING:

Exception Handling: Concepts of exception handling, Types of exceptions, Built-in exceptions, Usage of try, catch, throw, throws and finally keywords.

Multithreading: Concepts of Thread, Thread priorities, multithreading, Daemon thread, Synchronization.

UNIT-2 **10L+0T+8P=18 Hours**

AWT, APPLETS AND GUI PROGRAMMING WITH SWING:

AWT and Applets: Concepts of applets, differences between applets and applications, life cycle of applet, AWT, AWT Hierarchy, AWT Controls.

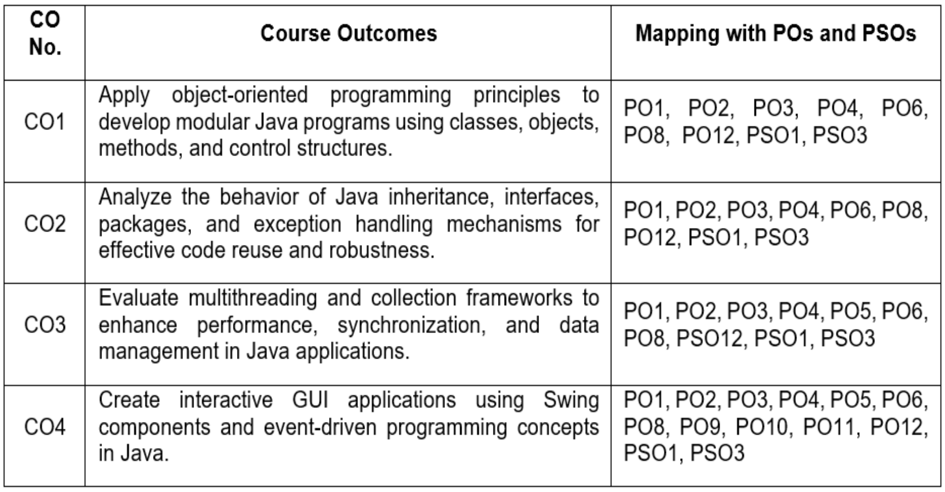
Exploring Swing Controls: JLabel JText Field, JButton, JCheckBox, JRadioButton, JTabbed Pane, JList, JCombo Box.

PRACTICES:

- Implementing Exception handling.
- Implement java program which accepts withdraw amount from the user and throws an exception “In Sufficient Funds” when withdraw amount more than available amount.
- Creating Thread.
- Implementing multithreading.
- Create three threads and that displays “good morning”, for every one second ,”hello” for every 2 seconds and “welcome” for every 3 seconds by using extending Thread class.
- Creating simple Applet.
- Develop an Applet program to accept two numbers from user and output the sum, difference in the respective text boxes.
- Implementing JLabel and JText Field.
- Implementing JButton and JRadioButton.
- Design student registration form using Swing Controls. The form which having the following fields and button SAVE a. Form Fields are: Name, RNO, Mailid, Gender, Branch, Address.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply object-oriented programming principles to develop modular Java programs using classes, objects, methods, and control structures.	PO1, PO2, PO3, PO4, PO6, PO8, PO12, PSO1, PSO3
CO2	Analyze the behavior of Java inheritance, interfaces, packages, and exception handling mechanisms for effective code reuse and robustness.	PO1, PO2, PO3, PO4, PO6, PO8, PO12, PSO1, PSO3
CO3	Evaluate multithreading and collection frameworks to enhance performance, synchronization, and data management in Java applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PSO12, PSO1, PSO3
CO4	Create interactive GUI applications using Swing components and event-driven programming concepts in Java.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3

TEXT BOOKS:

1. Herbert Scheldt, “Java the complete reference”, 12th Edition, McGraw Hill, Education, 2021.
2. T. Budd, “Understanding Object-Oriented Programming with Java”, Updated Edition, Pearson Education, 2000.

REFERENCE BOOKS:

1. J. Nino and F.A. Hosch, “An Introduction to programming and OO design using Java”, 3rd Edition, John Wiley & sons, 2008.
2. P. Radha Krishna, “Object Oriented Programming through Java”, 1st Edition, Universities Press, 2007.
3. R. A. Johnson, “Java Programming and Object Oriented Application Development”, 1st Edition, Cengage Learning, 2006.

Source: https://www.geeksforgeeks.org/best-way-to-start-with-competitive-programming-geeksforgeeks-cp-live-course/

22TP203 ADVANCED CODING COMPETENCY

Hours Per Week :

L	T	P	C
-	-	2	1

PREREQUISITE KNOWLEDGE: Programming in C, Data Structures.

COURSE DESCRIPTION AND OBJECTIVES:

This course helps to understand the impact of the choice of data structures and design strategies to solve the problem in an efficient manner. This course also provides the understanding of advanced graph applications and also throw light in tractable intractable problems.

MODULE-1

UNIT-1 **0L+0T+8P =8 Hours**

STACKS, QUEUES AND SINGLE LINKED LISTS:

PRACTICES:

Problems On Stacks & Queues

- Check if given stack of integers are consecutive or not (could be ascending or descending).
- Find the maximum sum in a sliding window using queues.
- Given a queue of integers, rearrange the elements by interleaving the first half with the second half.
- Given an integer k and a queue of integers, reverse the order of the first k elements of the queue.
- Given a maze in the form of a rectangular matrix filled with O, X or M where O represents an open cell, X represents a blocked cell and M represents landmines, find the shortest distance of every open cell in the maze from its nearest mine.
- For a given parenthesis expression, check whether it is balanced parenthesis or not.
- Reverse a number using stack.
- You are given a string s consisting of lowercase English letters. A duplicate removal consists of choosing two adjacent and equal letters and removing them. We repeatedly make duplicate removals on s until we no longer can.
- Find first Unique character in a string (Queue).
- Implement Tower of Hanoi problem.

Problems On Linked Lists

- Given a random pointer to a random node in a singly linked list, clone the list.
- Given a list rotate the list to the right by k places.
- Remove duplicates from a sorted list.
- Find fractional node in a singly linked list.
- Sort a linked list using constant space complexity.
- Delete a node in start, middle, end of Singly linked list.
- Add a node in start, middle, end of Singly linked list.
- Find whether given single linked list is circular or not.
- Arrange a singly linked list in Descending order.
- Addition of two numbers using Singly Linked List.

UNIT-2 **0L+0T+8P =8 Hours**

DOUBLY LINKED LISTS, CIRCULAR LINKED LISTS:

PRACTICES:

Problems on Double Linked Lists and Circular Linked Lists

- Implement a clockwise rotation of a doubly linked list by N places.
- Count triplets in a sorted doubly linked list whose product is equal to a given value x.
- Find the product of all prime nodes in a doubly linked list.
- Find the count of common nodes in two doubly linked lists.
- Find pairs with given product in a sorted doubly linked list.
- Delete all the even nodes of a circular singly linked list.
- Count nodes in a circular linked list.
- Delete all prime nodes from a circular singly linked list.
- Exchange first and last nodes in a circular linked list.
- Reverse a doubly circular linked list.
- Linear search using a stack of incomplete sub problems.
- 1 2 3 4 5 6 in stack S is push X is pop, SSSSXXSSSXXX.
- Recursively remove all adjacent duplicates.
- Check if a given singly linked list is a palindrome using stack.
- Convert a multilevel singly linked list to a singly linked list.
- Remove duplicates from an unsorted doubly linked list.
- Sort a doubly linked list using insertion sort.
- Check if a doubly linked list of characters is palindrome or not.
- Swap Kth node from beginning with Kth node from end in a Double Linked List.
- Convert a Binary Tree into Double Linked List.

MODULE-2

UNIT-1 **0L+0T+8P =8 Hours**

TREES:

PRACTICES:

Problems on Trees

- Given a sorted doubly linked list, convert it into a balanced BST.
- Given a singly linked list with data in the ascending order, convert it into a height balanced BST.
- Print the leaf to root path for every leaf node in a binary tree.
- Write a function to implement the reversed level order traversal of a binary tree.
- Truncate a given binary tree to remove nodes that lie on a path having sum less than K.
- Find the vertical sum in a given binary tree.
- Delete minimum & Maximum element from a BST.
- Implement Inorder, preorder and postorder tree traversal techniques.
- Print Kth largest element in a BST.
- Implement Zig-Zag tree traversal.

SKILLS:

- ✓ *Experienced to Store data and various types of data to handle.*
- ✓ *Ordering and sorting of data.*
- ✓ *Indexing and Searching of required data from large data sequences.*
- ✓ *Exposed to various characteristics such as Linear or non-linear, Homogeneous or heterogeneous and Static and Dynamic.*

UNIT-2 **0L+0T+8P =8 Hours**

GRAPHS:

PRACTICES:

Problems on Graphs

- Given a directed acyclic graph, determine whether there is a path that visits every vertex exactly once.
- Reverse a directed graph such that each edge from v to w is replaced by an edge from w to v.
- Find the shortest path in a graph that visits each vertex at least once, starting and ending at the same vertex.
- Find the minimum number of throws required to win a snake and ladder game.
- Implement DFS of a Graph.
- Implement BFS of a Graph.
- Detect whether a cycle is present in an undirected graph.
- Detect cycle in a Directed Graph.
- Find Shortest Distance to goal node from root node in a graph.
- Find no. of nodes in Kth level of a Graph.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply operations of stacks, queues, linked lists (singly, doubly, circular), trees, and graphs to solve common algorithmic problems.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12, PSO1,PSO2
CO2	Analyze the efficiency of data structure operations including insertion, deletion, traversal, and searching across stacks, queues, all types of linked lists, trees, and graphs.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12,PSO1,PSO2
CO3	Evaluate appropriate data structures (stack, queue, linked list, tree, graph) for solving problems involving recursion, backtracking, priority processing, and hierarchical or networked data.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12, PSO1,PSO2
CO4	Create optimized, modular, and reusable code for complex real-world problems using stacks, queues, linked lists, trees, and graphs that are frequently asked in technical interviews.	PO1, PO2, PO3,PO4, PO5, PO9, PO10, PO12, PSO1, PSO2

TEXT BOOKS:

1. Reema Thareja, "Data Structures Using C", 2nd Edition, Oxford University Press, 2014.
2. Seymour Lipschutz, "Data Structures with C", 1st Edition, McGraw Hill Education, 2017.

REFERENCE BOOKS:

1. Ellis Horowitz and Sartaj Sahni, "Fundamentals of Data Structures", illustrated edition, Computer Science Press, 2006.
2. Richard F. Gilberg and Behrouz A. Forouzan, "Data Structures: A Pseudocode Approach with C", 2nd Edition, CENAGE Learning, 2005.
3. R G Dromey and Pearson, "How to solve it by Computer", 2nd edition, Impression edition, 1998.

22TP204 PROFESSIONAL COMMUNICATION LABORATORY

Source: https://www.coursera.org/specializations/improve-english

Hours Per Week :

L	T	P	C
-	-	2	1

PREREQUISITE KNOWLEDGE: High School-level English.

COURSE DESCRIPTION AND OBJECTIVES:

To improve the overall professional communication skills (LSRW) of students and prepare them for their profession as engineers and managers. To provide them exposure to conventions of corporate communication and training them on how to function in the business world.

MODULE-1

UNIT-1 **0L+0T+8P=8 Hours**

BASICS OF BUSINESS WRITING SKILLS, PRACTICING BUSINESS CORRESPONDENCE AND REPORT WRITING:

Business English Vocabulary: Glossary of most commonly used words (formal and informal usage).

Elements of Technical Writing: Sentence structure, reducing verbosity, arranging ideas logically, building coherence, cohesive devices and transitional words.

Mechanics of Writing: Elementary rules of grammar, choice of diction, elementary principles of composition, matters of form, punctuation, conventions of business communication, language and professional tone, code of conduct (not sending illegal, offensive, disparaging personal remarks or comments) in written business communication.

Business Correspondence: E-mail: nature and scope, e-mail etiquette, clear call for action, common errors in composing e-mails, office communication such as meeting agenda and notice, circular and memo.

Letter-Writing: Formal and informal letters, structure of formal letters, expressions of salutations, different types of letters [such as sales letter, complaint letter, response to the complaint letter (dispute resolution), letter of permission, letter of enquiring, claim letter – letter of apology etc], introductory and concluding paragraphs and clear call for action.

Professional Proposal/Report: Differentiating proposals and reports, Drafting formal business proposals, types of reports such as factual reports, feasibility reports and survey reports, parts of a report (such as title page, declaration, acknowledgements, table of contents, abstract, introduction, findings, conclusion and recommendations).

New Age Corporate Communication Media: Importance of social media communication and Etiquettes, form and structure, sharing texts through Twitter, Whatsapp, instgram etc.

UNIT-2 **0L+0T+8P=8 Hours**

PRACTICING COMMUNICATIVE LANGUAGE IN VARIOUS PROFESSIONAL CONTEXTS:

Speaking: Speaking in business context, assertiveness, politeness, making requests, queries and questions, negotiations, asking for information, offering suggestions, conflict resolution, contacting clients, initiating, addressing delegates (in public), delivering the presentation effectively, telephone etiquettes, delivering seminar/proposal/report effectively, team meeting etiquettes (face to face and conference call), making effective one minute presentations(JAM) and participating in Group Discussions.

PRACTICES:

- Basic grammar practice, framing paragraphs on topics allocated, paraphrasing an article or a video in your own words, finding topic sentences in newspaper articles, finding out new words from a professional viewpoint and understanding the meaning and its usage.

- Perusing samples of well-prepared business emails, memo, letter writing and short proposals and reports, students will draft business correspondence writing tasks and different proposals/ reports on topics assigned.
- Watching videos/listening to audios of business presentations, classroom activities of team and individual presentations, using PPTs, mock exercises for BEC speaking, agreeing, disagreeing politely, developing content, extended speaking in Group Discussion(s).

MODULE-2

UNIT-1 **0L+0T+8P=8 Hours**

READING AND COMPREHENDING BUSINESS DOCUMENTS:

Reading: Reading and comprehending business documents, learning business register, regularizing the habit of reading business news, suitable vocabulary, skimming and scanning a text for effective and speedy reading and dealing with ideas from different sectors of corporate world in different business contexts.

UNIT-2 **0L+0T+8P=8 Hours**

IMPARTING AND PRACTICING LISTENING SKILLS:

Listening: Specific information in business context, listening to telephonic conversations / messages and understanding the correct intended meaning, understanding the questions asked in interviews or in professional settings, summarizing speaker’s opinion or suggestion, enable active listening.

PRACTICES:

- Hand-outs; matching the statements with texts, finding missing appropriate sentence in the text from multiple choices, using right vocabulary as per the given context and editing a paragraph.
- Working out BEC/TOEFL/IELTS listening exercises with hand-outs; matching the statements with texts, finding missing appropriate sentence in the text from multiple choice- multiple choices, using right vocabulary in context-editing a paragraph, listening to a long conversation such as an interview and answer MCQ s based upon listening.

SKILLS:

- ✓ *To enhance listening and spoken abilities of students needed for professional and social success in interpersonal situations, group interactions, and personal and professional presentations.*
- ✓ *Understand and practice specific functions and vocabulary in a business context.*
- ✓ *Produce short business reports, proposals and correspondence.*
- ✓ *Write various business documents through reading techniques.*

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the principles of business writing, email etiquette, and formal letter drafting to communicate clearly and professionally in corporate scenarios..	PO5, PO6, PO8, PO9, PO10, PO12, PSO1
CO2	Analyze and prepare structured business reports, proposals, and documents using standard formats, tone, grammar, and mechanics suited for workplace communication.	PO5, PO6, PO8, PO9, PO10, PO12, PSO1
CO3	Evaluate and participate in professional spoken interactions, including group discussions, telephonic calls, client meetings, and public presentations using appropriate tone, assertiveness, and etiquettes.	PO5, PO6, PO8, PO9, PO10, PO12, PSO1
CO4	Create context-appropriate responses after reading and listening to various business texts and conversations, demonstrating comprehension, speed reading, and meaningful interpretation.	PO5, PO6, PO8, PO9, PO10, PO12, PSO1

TEXT BOOK:

1. S. Schnurr, “Exploring Professional Communication: Language in Action”, London: Routledge, 2013

REFERENCE BOOKS:

1. Brook Hart Guy, “Cambridge English Business Bench Mark: Upper Intermediate”, 2nd Edition: CUP, 2014.
2. Cambridge University Publication, “Cambridge: BEC VANTAGE Practice Papers”, CUP, 2002.
3. J. Seely, “The Oxford Guide to Effective Writing and Speaking”, Oxford University Press, 2005.

22IT204 DESIGN AND ANALYSIS OF ALGORITHMS

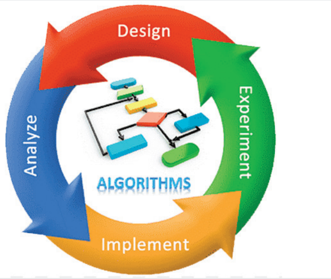


IMAGE SOURCE: https://e7.pngegg.com/pngimages/328/186/png-clipart-introduction-to-algorithms-algorithms-design-and-analysis-analysis-of-algorithms-algorithm-design-design-text-development-thumbnail.png

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Data Structures and Algorithms

COURSE DESCRIPTION AND OBJECTIVES:

Algorithm design and analysis provide the theoretical backbone of computer science and are a must in the daily work of the successful programmer. The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able to develop their own versions for a given computational task and to compare and contrast their performance.

MODULE-1

UNIT-1 **9L+0T+6P=15 Hours**

BASICS OF ALGORITHMS:

Introduction: Algorithm; Pseudo-code for expressing algorithms; Performance analysis; Asymptotic notations; Recursive Algorithms; Randomized Algorithms; Time complexity by Master's Theorem; Disjoint sets; Graph representations; Connected components and Bi-connected components.

Divide & Conquer and Greedy Methods: Divide and conquer general method; Greedy general method.

UNIT-2 **15L+0T+10P=25 Hours**

APPLICATIONS OF DIVIDE AND CONQUER AND GREEDY METHODS:

Applications: Towers of Hanoi Problem; Binary search; Quick sort; Merge sort; and Stassen's matrix multiplication; Max-min problem; Job sequencing with deadlines; Knapsack problem; Minimum cost spanning trees.

PRACTICES:

- State Towers of Hanoi problem. Implement Towers of Hanoi Problem. Repeat the experiment for different values of n (number of disks) by taking n value from a sorted list or array. **(RECURSIVE ALGORITHMS)**
- Implement binary search algorithm to find the existence of a particular element in the list or array, repeat the experiment for different values of n (number of elements in the given list or array). **(DIVIDE AND CONQUER)**
- Sort the given list of elements using the following methods, repeat the experiments for different values of n (number of elements in the given list or array) **(DIVIDE AND CONQUER)**
  - o Merge sort
  - o Quick sort
- Implement divide and conquer strategy to find the max and min elements of given list or array. Repeat the experiment for different values of n (number of elements in the given list or array). **(DIVIDE AND CONQUER)**
- Implement Strassen's matrix multiplication to multiply two 2X2 matrices. **(DIVIDE AND CONQUER)**
- State the problem of Job sequencing with deadlines and write a program to solve it using Greedy method. **(GREEDY METHOD)**
- State Knapsack problem and write a program to solve the knapsack problem using Greedy approach. **(GREEDY METHOD)**

SKILLS:

- ✓ *Students are able to determine suitable algorithm design strategy to develop an algorithm for solving the given novel problem.*
- ✓ *Students can apply various algorithm design paradigms (divide-and-conquer, greedy method, dynamic-programming and backtracking and branch and bound) to solve the given problem.*
- ✓ *Students are able to analyze best, average and worst-case running times of algorithms using asymptotic analysis.*
- ✓ *Student can develop new algorithms for solving the real time applications using different algorithm design strategies.*

MODULE-2

UNIT-1 **6L+0T+4P=10 Hours**

DYNAMIC PROGRAMMING, BACK TRACKING AND BRANCH&BOUND METHODS:

Introduction: Dynamic programming general method; back tracking general method; branch and bound general method; introduction to the applications of dynamic programming, back tracking and branch and bound.

UNIT-2 **18L+0T+12P=30 Hours**

APPLICATIONS OF DYNAMIC PROGRAMMING, BACK TRACKING AND BRANCH & BOUND METHODS:

Applications: Optimal binary search trees; Matrix chain multiplication; 0/1 knapsack problem; All pairs shortest path problem; Travelling sales person problem; Reliability design problem; n-queen problem; Sum of subsets problem; Graph colouring; Hamiltonian cycles; Traveling sales person problem; 0/1 knapsack problem; Job assignment problem.

PRACTICES:

- State all pairs shortest path problem and write a program to solve it using dynamic programming. **(DYNAMIC PROGRAMMING)**
- State Travelling sales person problem. Find the tour of minimum cost of travelling sales person by using dynamic programming. **(DYNAMIC PROGRAMMING)**
- State the problem of Matrix chain multiplication and write a program to find the cost of optimal order of matrix chain multiplication problem using dynamic programming. **(DYNAMIC PROGRAMMING)**
- State n-queens' problem and write a program to solve the same using backtracking approach. **(BACK TRACKING METHOD)**
- State the problem of sum of subsets and write a program to solve it using backtracking approach. **(BACK TRACKING METHOD)**
- Distinguish Fractional Knapsack and 0/1 Knapsack problem. Write a program to solve knapsack problem using Branch and Bound. **(BRANCH AND BOUND METHOD)**

COURSE OUTCOMES:

Upon successful completion of this course, students will have to ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply various algorithm design paradigms (divide-and-conquer, greedy method, dynamic-programming and backtracking and branch and bound) to solve the given problem.	PO1,PO2PO3,PO4,PO5 PO12, PSO1
CO2	Analyze best, average and worst-case running times of algorithms using asymptotic analysis.	PO1,PO2PO3,PO4,PO5 PO12, PSO1
CO3	Determine suitability of algorithm design strategy to develop an algorithm for solving the given novel problem	PO1,PO2PO3,PO4,PO5 PO12, PSO1
CO4	Develop new algorithms for solving the real time applications using different algorithm design strategies.	PO1,PO2PO3,PO4,PO5 PO12, PSO1

TEXT BOOK:

1. Ellis Horowitz, Satraj Sahni and Rajasekharam, "Fundamentals of Computer Algorithms", 2nd Edition, Galgotia publications, 2006.

REFERENCE BOOKS:

1. Anony Levitin, "Introduction to Design and Analysis of Algorithms", 3rd Edition, Pearson Education, 2016.
2. Donald E.knuth, "The Art of Computer Programming", Volume 3, 2nd Edition, Addison Wesley Longman Inc,1998.
3. Ronald L. Graham, Donald E. Knuth and Oren Patashnik, "Concrete Mathematics", 2nd Edition, Addison-Wesley Publishing Company,1998.

22IT205 OPERATING SYSTEMS

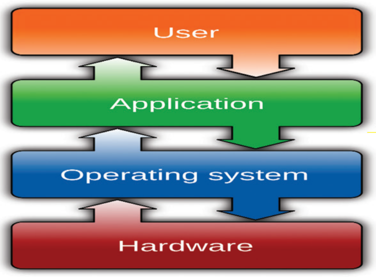


Image Citation: https://upload.wikimedia.org/wikipedia/commons/thumb/e/e1/Operating_system_placement.svg/250px-Operating_system_placement.svg.png

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Data structures, Programming, Computer Organization

COURSE DESCRIPTION AND OBJECTIVES:

This course aims at concepts and principles of Operating Systems, its overall responsibility in acting as an interface between the system's hardware components and the user/application software. Further, it also helps students to understand the different scheduling policies, process synchronization mechanisms, deadlock handling mechanisms, memory management techniques and file management system.

MODULE-1

UNIT-1 **12L+0T+6P=18 Hours**

PROCESS MANAGEMENT:

Introduction and Process: What Operating System do, Operating System structure, Process concept - overview, process scheduling, operations on process, Threads, Inter-process communication, Process scheduling - scheduling criteria, scheduling algorithms, Multiple-Processor scheduling.

UNIT-2 **12L+0T+10P=22 Hours**

PROCESS SYNCHRONIZATION AND DEADLOCK:

Process Synchronization: The critical-section problem, Peterson's solution, Synchronization hardware, Semaphores; Monitors, Classical problems of synchronization.

Deadlocks: Deadlock characterization, Methods of handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection and recovery.

PRACTICES:

- Process creation, termination and managing the processes.
- Thread creation, termination and managing the threads.
- Simulation of CPU Scheduling algorithms.
- Simulating the solutions of process synchronization.
- Simulating banker's algorithm for given data and reporting the findings.

MODULE-2

UNIT-1 **12L+0T+6P=18 Hours**

MEMROY MANAGEMENT:

Memory Management: Continuous memory allocation, Paging, Structure of the page table, Segmentation, Demand paging, Page replacement algorithms.

UNIT-2 **12L+0T+10P=22 Hours**

FILE SYSTEM AND SECONDARY MEMORY MANAGEMENT:

Mass Storage: Overview of mass-storage structure, disk structure, disk scheduling

File Management: File Systems - file concept, access methods, directory structure, file system mounting, file sharing protection; File-system structure, file system implementation, directory implementation, allocation methods, free space management.

SKILLS:

- ✓ *Mange open source operating systems like Ubuntu, Fedora etc.*
- ✓ *Understand the concepts of process scheduling and files systems.*
- ✓ *Analyze the various algorithms used for memory management.*
- ✓ *Identify various disk scheduling methodologies.*

PRACTICES:

- Simulating the partition allocation methods.
- Simulate the memory management techniques like paging, segmentation.
- File, process, and disk management commands in linux/unix.
- Simulate the file manipulation commands in programming.
- Design and develop free space management techniques.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply operating system concepts related to processes, scheduling, and inter-process communication to manage system resources efficiently.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12, PSO1,PSO3
CO2	Analyze synchronization mechanisms and deadlock handling strategies to improve process coordination and safety.	PO1, PO2, PO3, PO4, PO5, PO6, PO9,PO10, PO12, PSO1, PSO3
CO3	Evaluate memory and storage management techniques, including paging, segmentation, and file systems for performance optimization.	PO1, PO2, PO3, PO4, PO5, PO6, PO9, PO10,PO12, PSO1, PSO3
CO4	Create simple scheduling or memory management simulations to demonstrate real-time operating system behavior and timing constraints.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3

TEXT BOOK:

1. Abraham Silberschatz Peter B. Galvin and Greg Gagne, “Operating System Concepts”, 9th Edition, Wiley, 2013.

REFERENCE BOOKS:

1. Garry. J. Nutt, “Operating Systems: A Modern Perspective”, 3rd Edition, Addison-Wesley, 2016.
2. Andrew S. Tanenbaum and Herbert Bros, “Modern Operating Systems”, 4th Edition, Pearson, 2015.
3. William Stallings, “Operating Systems: Internals and Design Principles”, 7th Edition, Prentice Hall of India, 2013.
4. Dhananjay M. Dhamdhere, “Operating Systems: A Concept-Based Approach”, 3rd Edition, Tata McGraw-Hill, 2017.

22IT206 PYTHON PROGRAMMING

Image source: https://webnotics.solutions/python-programming/

Hours Per Week :

L	T	P	C
2	-	2	3

PREREQUISITE KNOWLEDGE: Computer Programming

COURSE DESCRIPTION AND OBJECTIVES:

This course offers sufficient knowledge required to understand the fundamental concepts of Python programming language. This course enables students to choose appropriate data structures like lists, dictionaries, tuples, sets, strings for the given problem. In addition, the students will be able to create reliable, modular and reusable applications using Object- Oriented Programming approaches.

MODULE-1

UNIT-1 **6L+0T+6P=12 Hours**

PYTHON BASICS:

Python Installation and Working of it, get familiar with python variables and data types, Operator understanding and its usage, detail study of python blocks, Hands on with conditional blocks using if, else and elif

UNIT-2 **10L+0T+10P=20 Hours**

WORKING WITH VARIETIES OF DATA:

Hands on string handling and looping with range, list, Tuples, Sets and dictionaries. hands on to organize python code with function, modular approach in python.

PRACTICES:

- Installation of python and relevant packages in windows.
- Installation of python and relevant packages in Linux.
- Practice Execution of python statements in REPL(shell).
- Implement a python program to display all the python keywords and display each of them in separate lines.
- Develop a python program to read two integers and perform all possible arithmetic operations on those two numbers.
- Develop a program to accept three numbers as command line arguments and find biggest, smallest and average of those three numbers.
- Implement a python program to find first n Prime Numbers.
- Implement a program that prints the decimal equivalents of 1/2, 1/3, 1/4, . . . ,1/n.
- Implement a python program to read n and find sum of even and odd numbers.
- Write python code to achieve the following
  - o to remove vowels in the given string using control transfer statements.
  - o to count number of uppercase and lowercase letters in the given string.
  - o to remove all punctuation characters from given string.
- Implement python code to illustrate the following on Lists and Tuples
  - o Creation b) Accessing elements c) apply operators d) Usage of different methods
- Implement python code to illustrate the following on Sets and Dictionary
  - o Creation b) Accessing elements c) apply operators d) Usage of different methods
- a) Implement python code to illustrate the following

SKILLS:

- ✓ *Identify suitable data types and data structures required for an application*
- ✓ *Design structured and Object oriented programming solutions*
- ✓ *Design reliable applications for a given problem.*

- Positional arguments ii) Keyword arguments iii) Default arguments iv) Variable length arguments
- Implement a function to find nth Fibonacci number.
- Develop a recursive function to find the factorial of a given number.
- Implement function to compute GCD, LCM of two numbers (use Lambda function).

MODULE-2

UNIT-1 **8L+0T+8P=16 Hours**

EXCEPTION AND FILE HANDLING:

Handling and helping file operations, coding with the exceptional handling

UNIT-2 **8L+0T+8P=16 Hours**

OBJECT ORIENTED PROGRAMMING:

Object-Oriented Programming, Classes and working with instances, Method overloading, Polymorphism

PRACTICES:

- Develop a python code to handle the following built-in exceptions
  - o ValueError ii) ZeroDivisionError iii) TypeError iv) NameErrror
- Implement python code to handle multiple exceptions.
- Implement Python code to raise an exception.
- Implement python code to read contents of a file and write the contents to another file.
- Create a class called Student and perform operations such as display, Calculate percentage, add, delete and modify student data.
- Design python code to depict the following oops concepts: i) Datahiding ii) Inheritance iii) Overriding
- Develop python code to calculate the following statistical parameters using python 'numpy'.
- Mean b) Hormonic Mean c) Meadian d) Mode e) Standard Deviation f) Variance g) Percentile
- Design python code to illustrate the following plots using 'matplotlib' package
- Line plot b) Bar plot c) Histogram d) Scatter Plot
- Implement python program for the following problems on Pandas DataFrame
- Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels. Sample Python dictionary data and list labels:
- exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
  'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
- 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
  'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
- labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
- Write a Pandas program to select the 'name' and 'score' columns from the following DataFrame.
- Write a Pandas program to select the specified columns and rows from a given data frame.
- Write a Pandas program to select the rows where the number of attempts in the examination is greater than 2.
- Write a Pandas program to count the number of rows and columns of a DataFrame.
- Write a Pandas program to change the name 'James' to 'Adhvik' in name column of the DataFrame.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply Python control structures, and data types to develop basic programs and solve simple computational problems.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO11, PO12, PSO1,PSO3
CO2	Analyze built-in data structures, functions, and file handling features to organize and manipulate data effectively.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO11,PO12, PSO1, PSO3
CO3	Evaluate exception handling, modular programming, and object-oriented principles to enhance the robustness, readability, and maintainability of Python applications.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO11, PO12,PSO1, PSO3
CO4	Create object-oriented Python applications by designing classes, using inheritance, and implementing data encapsulation.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO11,PO12, PSO1, PSO3

TEXT BOOK:

1. Reema Thareja, “Python Programming: Using Problem Solving Approach”, 1st Edition, Oxford University Press, 2017.

REFERENCE BOOKS:

1. Eric Matthes, “Python Crash Course: A Hands-On, Project-Based Introduction to Programming”, 2ndEdition, No Starch Press, 2019.
2. John V. Guttag, “Introduction to Computation and Programming Using Python”, 3rd Edition, The MIT Press, 2021
3. Allen B. Downey,” Think Python”, 2nd edition, O’rielly publishing, 2015.
4. Vamsi Kurama, “Python Programming: A Modern Approach”, 1st Edition, Pearson Publishers, 2018.

III

YEAR

B.Tech.

INFORMATION TECHNOLOGY

I SEMESTER

22TP301	-	Soft Skills Lab
22IT301	-	Computer Networks
22IT302	-	Data Mining Techniques
22IT303	-	Software Engineering
	-	Department Elective – 2
	-	Open Elective – 2
22IT304	-	Inter-Disciplinary Project – Phase I
22IT305	-	Industry interface course (Modular course)
		NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication
		Minor / Honors – 2

II SEMESTER

22TP302	-	Quantitative aptitude & Logical reasoning
22IT306	-	Cloud Computing
22IT307	-	Machine Learning
	-	Department Elective – 3
	-	Department Elective – 4
	-	Open Elective – 3
22IT308	-	Inter-Disciplinary Project – Phase II
	-	Minor / Honors – 3

COURSE CONTENTS

I SEM & II SEM

22TP301 SOFT SKILLS LABORATORY

Source: https://choosework.ssa.gov/blog/2019-07-23-soft-skills-an-intro-to-effective-communication

Hours Per Week :

L	T	P	C
-	-	2	1

PREREQUISITE KNOWLEDGE: Grasp on their own academic achievements.

COURSE DESCRIPTION AND OBJECTIVES:

To impart employability skills like resume preparation and facing interviews. To enable trainees to develop interpersonal and leadership skills and to train them on work place skills like making presentations, participating in group discussions etc.

MODULE-1

UNIT-1 **0L+0T+8P=8 Hours**

PERSONALITY DEVELOPMENT:

Soft Skills: Need for soft skills, professionalism, employability skills; Communication: Need for effective communication - the process of communication, levels of communication, flow of communication, choice of diction and style with reference to setting (formal, semi-formal or informal); communication networks, barriers to communication, miscommunication, noise and ways to overcome the barriers; Career Planning: Job vs. career, SWOT analysis.

UNIT-2 **0L+0T+8P=8 Hours**

LANGUAGE AND VOCABULARY:

Vocabulary Building: Word etymology, roots, prefixes & suffixes, synonyms & antonyms, collocations, one-word substitutes, analogies, idioms and phrases, contextual guessing of unfamiliar words, task-oriented learning; Reflection of language on Personality, Gender sensitive language in MNCs, Mind your language, Seven essential skills for a team player; attentive listening, intelligent questioning, gently persuading, respecting other’s views, assisting others, sharing, participating actively.

PRACTICES:

- Self-Introduction.
- Personal and Academic SWOC.
- Johari Window.
- Giving and taking opinions of Self Vs others and assessing oneself.
- Goal setting.
- Short, Mid and Long Term goals planning the semester.
- Time management: four quadrant system.
- Stephen Covey Time Management Matrix planning a semester.
- Stress-management.
- Questionnaire to assess level of stress.
- 50 words towards resume preparation and interviews.
- Newly coined words.
- Gender sensitive words and Words acceptable in Indian context and objectionable international context.

MODULE-2

UNIT-1 **0L+0T+8P=8 Hours**

LANGUAGE IN ACTION:

Functional English: Situational dialogues, Role plays (including small talk); Group Discussion: Articulation and flow of oral presentation, dynamics of group discussion, intervention, summarizing and conclusion, voice modulation, content generation, Key Word Approach (KWA), Social, Political, Economic, Legal and Technical Approach (SPELT), View Point of Affected Part (VAP), language relevance, fluency and coherence – 11th and 12th weeks; Resume preparation: Structure and presentation, defining career objective, projecting one's strengths and skill-sets, summarizing, formats and styles and covering letter-Statement of Purpose.

UNIT-2 **0L+0T+8P=8 Hours**

PREPARING FOR PRESENTATIONS AND INTERVIEWS:

Facing Interviews: Interview process, understanding employer expectations, pre-interview planning, opening strategies, impressive self-introduction, answering strategies, other critical aspects such as body language, grooming, other types of interviews such as stress-based interviews, tele- interviews, video interviews, frequently asked questions (FAQs) including behavioral and HR questions and the aspect looked at by corporate during interviews; Presentation Skills: Selection of a topic, preparing an abstract, gathering information, organizing the information, drafting the paper, citing reference sources – writing striking introductions, discussing the methodology used, developing the argument, presentation style, language, presenting the paper and spontaneously answering audience questions.

PRACTICES:

- Opening and closing a telephonic conversation.
- Making an appointment.
- Making a query.
- Offering/Passing on information.
- Communicating with superiors.
- Expressing agreement/objection.
- Opening bank account (combination of prepared and impromptu situations given to each student).
- Group Discussions on various topics.
- Preparing SoP and Resume.
- Mock interviews on the FAQs including feedback.
- Oral presentation with the help of technology (Preparing PPT and presenting).

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply communication skills effectively in both verbal and written formats suitable for professional and workplace scenarios.	PO5, PO8, PO9, PO10, PO12
CO2	Analyze individual strengths, weaknesses, and time-management patterns to plan and manage career goals effectively.	PO5, PO8, PO9, PO10, PO12
CO3	Evaluate perspectives in group discussions, interviews, and role plays to engage in collaborative and professional interactions.	PO5, PO8, PO9, PO10, PO12
CO4	Create structured professional documents such as resumes, SOPs, emails, and technical presentations demonstrating clarity, coherence, and professionalism.	PO5, PO8, PO9, PO10, PO12

TEXT BOOKS:

1. Adrian Furnham, "Personality and intelligence at work", Psychology Press, 2008.
2. S. P. Dhanvel, "English and Soft skills", Orient Blackswan, 2011.

REFERENCE BOOKS:

1. Edward Holffman, "Ace the corporate personality", McGraw Hill, 2001.
2. John Adair Kegan Page, "Leadership for innovation", Kogan, 2007.

SKILLS:

- ✓ *Balance social and emotional intelligence quotients though SWOC, JOHARI etc. activities.*
- ✓ *Prepare tailor made resume and face various job interviews with enriched personality traits.*
- ✓ *Career planning with clear personal and professional goals.*
- ✓ *Solve personal and professional life hiccups with confidence and maturity.*

22IT301 COMPUTER NETWORKS

Image source: https://www.javatpoint.com/computer-network-tutorial

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Data structures and Computer programming

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on imparting knowledge about various protocols involved in LANs and WANs. In addition, it gives good foundation on different protocols such as data link protocols, internet protocols and transport protocols present in the respective layers of computer networks.

MODULE-1

UNIT-1 **12L+0T+6P=18 Hours**

PHYSICAL LAYER AND DATA LINK LAYER:

Uses of computer networks, Network hardware, Network software, Reference Models, Guided transmission media, Data link layer design issues, Elementary data link protocols, The channel allocation problem.

UNIT-2 **12L+0T+10P=22 Hours**

MAC LAYER:

Medium allocation methods, Error detection and correction, Sliding window protocols, Example data link protocols, Multiple access protocols.

PRACTICES:

- Bit stuffing and byte stuffing.
- Error detection and correction.
- Performance calculation in MCA protocols.
- Performance evaluation in sliding window protocol.
- Study on physical addressing.

MODULE-2

UNIT-1 **10L+0T+6P=16 Hours**

NETWORK LAYER:

Network layer design issues, Routing algorithms - optimality principle, shortest path routing, flooding, distance vector routing, link state routing and hierarchical routing; The network layer in the internet - IP addresses, Services provided to the upper layers.

UNIT-2 **14L+0T+10P=24 Hours**

TRANSPORT LAYER AND APPLICATION LAYER:

Elements of transport protocols, Congestion control algorithms, QoS improving techniques, leaky bucket and token bucket algorithms, Internetworking, IPv4, IPv6, IMCP, ARP, DHCP and mobile IP, Congestion Control, UDP and TCP, Performance Issues, The world wide web, Domain name system, E-mail, Streaming Audio and Video, Content Delivery.

SKILLS:

- ✓ *Implement Local Area Networks with different topologies.*
- ✓ *Simulate various routing protocols.*
- ✓ *Network troubleshooting.*
- ✓ *Work on various network devices to establish a Network for Institute.*
- ✓ *Classify different classes of IP addresses.*

PRACTICES:

- Logical addressing division.
- Performance evaluation routing and congestion control algorithms.
- TCP/IP programming.
- UCP/IP programming.
- Configuration of email system.

COURSE OUTCOMES:

Upon successful completion of this course, students will have to ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply fundamental concepts of data communication, network models, and signal transmission to understand network architecture and standards.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO12, PSO1, PSO2
CO2	Analyze error control, multiple access protocols, and Ethernet standards to ensure efficient and reliable data link layer communication.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO12, PSO1, PSO2
CO3	Evaluate network, transport, and routing protocols including IPv4/IPv6, TCP, UDP, and QoS mechanisms to optimize end-to-end communication.	PO1, PO2, PO5, PO6, PO8, PO12, PSO1, PSO2
CO4	Evaluate application-layer services such as DNS, email, and file transfer to understand their functionality and role in network communication.	PO1,PO2, PO5, PO6, PO8, PO12, PSO1, PSO2

TEXT BOOKS:

1. Andrew S.Tanenbaum, “Computer Networks”, 5th Edition. Pearson Education, 2014.

REFERENCE BOOKS:

1. Behrouz A. Forouzan, “Data communications and Networking”, 4th Edition, TMH, 2017.
2. William Stallings, “Data and Computer Communications”, 10th Edition, Pearson Education, 2017.

22IT302 DATA MINING TECHNIQUES

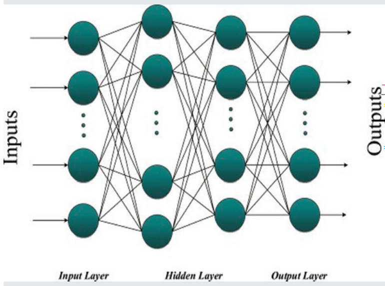


Image source: https://ars.els-cdn.com/content/image/3-s2.0-B9780128210925000061-f11-01-9780128210925.jpg

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Basics of statistics, Linear algebra

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the concepts, principles, methods and implementations of data mining subject, with a focus on 3 major data mining functions: 1. Association rule mining 2. Classification 3. Cluster Analysis. In the first part of the course student will learn terminology of data mining and different kinds of data pre-processing techniques. In the second part of the course student will learn major data mining techniques, their implementation and applications.

MODULE-1

UNIT-1 **10L+0T+6P=16 Hours**

INTRODUCTION TO DATA MINING:

Introduction to the need of data mining, Different kinds of data and patterns that can be mined, basic statistical descriptions of data, matrix representation of data, similarity and dissimilarity, need of the data pre-processing, handling missing values and noise in data, identifying redundant and correlated attributes, finding frequent item sets and generating association rules using Apriori and FP growth algorithms.

UNIT-2 **14L+0T+10P=24 Hours**

DATA HANDLING USING STATISTICAL METHODS:

Computing basic statistical descriptions of different kinds of data(mean, median, mode, variance, and standard deviation), distance and similarity metrics(Euclidean, Manhattan, cosine).Missing values handling methods, computing correlation of nominal and numerical data using Chi-square & Pearson method, computing Frequent patterns using Apriori and FP growth methods.

PRACTICES:

- Computing mean, median, mode, variance, Standard deviation measures.
- Matrix representations of data.
- Computing Euclidean, Manhattan, Cosine distance measures.
- Missing values handling methods.
- Chi-square method.
- Pearson method.
- Finding frequent patterns and association rules using Apriori algorithm.
- Finding frequent patterns and association rules using Frequent Pattern growth algorithm.

MODULE-2

UNIT-1 **10L+0T+6P=16 Hours**

CLASSIFICATION:

Introduction to classification and different approaches to perform classification. Concept of Information Gain, gain ratio and Gini Index, Bayes theorem and its terms, linearly separable and inseparable cases, concept of Neural Networks and working model of Multilayer Perceptron classifier, model evaluation techniques, classification through Encembling Techniques.

SKILLS:

- ✓ *Pre-processing the data.*
- ✓ *Understanding the statistics of data.*
- ✓ *Finding frequent patterns and forming association rules.*
- ✓ *Classification of Numerical data.*
- ✓ *Finding the clusters for the given un-labelled data.*

UNIT-2 **14L+0T+10P=24 Hours**

CLUSTERING:

Introduction to clustering and different approaches for clustering. K means and K medoid methods of partitioning based clustering, Agglomerative and Divisive methods of hierarchical clustering. Cluster evaluation methods.

PRACTICES:

- Decision tree classifier.
- Naïve Bayes classifier.
- KNN classifier.
- SVM classifier.
- Forward propagation of NN.
- Updating the weights for backward propagation of NN.
- K means and K medoid Clustering technique.
- Agglomerative and Divisive clustering technique.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply data mining techniques for data preprocessing, pattern discovery, classification, and clustering to solve real-world data problems.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1,PSO3
CO2	Analyze data characteristics, association patterns, and classification models to identify meaningful insights from complex datasets.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1,PSO3
CO3	Evaluate data mining algorithms and models using performance metrics, and select appropriate techniques for specific data mining tasks.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1,PSO3
CO4	Create data-driven solutions by integrating preprocessing, pattern mining, classification, and clustering methods to support intelligent decision-making.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1,PSO3

TEXT BOOKS:

1. Jiawei Han, Micheline Kamber and Jian Pei, “Data mining Concepts and Techniques”, 3rd edition, Morgan Kaufmann, 2012.

REFERENCE BOOKS:

1. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction to Data Mining”, 2nd edition, Pearson, 2018.
2. Jure Leskovec, Anand R aja raman and Jeffrey D Ullman, “Mining of Massive Datasets”, 5th edition, Stanford University, 2014.

22IT303 SOFTWARE ENGINEERING

Source: https://sites.google.com/site/softwengi/

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Data base management systems, Operating systems, Object oriented programming.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the concepts of software development life cycle, role of process models and methods to prepare software requirement specification document. In addition to that, it also imparts knowledge of design, development and testing of software. The objective of this course is to enable the student to develop defect free and efficient software as per the given requirements.

MODULE-1

UNIT-1 **14L+0T +4P=18 Hours**

INTRODUCTION TO SOFTWARE ENGINEERING: GENERIC VIEW OF PROCESS-PROCESS MODELS:

Introduction to Software Engineering: Software, Project, Software myths. Software Engineering - A layered technology A process framework, Software Development Life Cycle (SDLC), The Capability Maturity Model Integration (CMMI), Process Assessment. The Waterfall Model, Incremental Process Models, Evolutionary Process Models (Spiral and Prototype models).

UNIT-2 **10L+0T+12P=22 Hours**

AN AGILE VIEW OF PROCESS- REQUIREMENTS ENGINEERING- BUILDING THE ANALYSIS MODEL:

Agile Process Models: The Unified process, Extreme Programming, Scrum. Inception, Elicitation, Elaboration, Negotiation, Specification (SRS Document, IEEE Standards for SRS), Validation, Requirements management, Feasibility Study.

Data Modeling: Data objects, Attributes, Relationship, Cardinality and modality

Class based Modeling: Identify analysis classes, specify attributes and Define operations.

PRACTICES:

- Identifying the Requirements from Problem Statements. (Take any Real Time Problem).
- Estimation of Project Metrics.
- Modeling UML Use Case Diagrams and Capturing Use Case Scenarios.
- State chart and Activity Modeling.

MODULE-2

UNIT-1 **14L+0T+4P=18 Hours**

DESIGN ENGINEERING CREATING AN ARCHITECTURAL DESIGN- PRODUCT METRICS - QUALITY MANAGEMENT:

Design Concepts: The design model. Software architecture; Data design; Architectural styles and patterns; Architectural design. Golden rules; User interface analysis and design; Interface analysis; Interface design steps; Design evaluation.

SKILLS:

- ✓ *Define a process for developing/ completing different kinds of projects on time with expected quality.*
- ✓ *Understand the software requirements and find out various ways to gather them and specifying them.*
- ✓ *Analyze and model (Diagrammatical/ Representations) a software product.*
- ✓ *Design an effective, user-friendly interface for a given software product.*

Product Metrics: Metrics for analysis model; Metrics for design model; Architecture metrics and Object-oriented metrics; Metrics for source code; Metrics for testing; Metrics for maintenance.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software Quality Assurance, Configuration management, assessing and controlling software quality.

UNIT-2 **10L+0T+12P=22 Hours**

TESTING STRATEGIES - RISK MANAGEMENT:

Testing Strategies: A strategic approach to software testing; Test strategies for conventional software; Validation testing; System testing. Black-Box and White-Box testing.

Risk Management: Reactive vs Proactive risk strategies; Software risks; Risk identification; Risk projection; Risk refinement; RMMM; RMMM plan. Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software Quality Assurance, Configuration management, assessing and controlling software quality.

PRACTICES:

- Modeling UML Class Diagrams and Sequence diagrams.
- Estimation of Test Coverage Metrics and Structural Complexity.
- Design of Test Cases.

LIST OF PROJECTS:

Note: The student should take any THREE projects in the below and complete all practices

PROJECT -1

A POINT-OF-SALE (POS) SYSTEM: A POS system is a computerized application used to record sales and handle payments; it is typically used in a retail store, it includes hardware components such as a computer and bar code scanner, and software to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault tolerant; that is, even if remote services are temporarily unavailable they must still be of capturing sales and handling at least cash payments. A POS system must support multiple and varied client-side terminals and interfaces such as browser, PDAs, touch-screens.

PROJECT - 2

ONLINE BOOKSHOP EXAMPLE: Following the model of amazon.com or bn.com, design and implement an online bookstore.

PROJECT - 3

A SIMULATED COMPANY: Simulate a small manufacturing company. The resulting application will enable the user to take out a loan, purchase a machine, and over a series of monthly production runs, follow the performance of their company.

PROJECT - 4

A MULTI-THREADED AIRPORT SIMULATION: Simulate the operations in an airport. Your application should support multiple aircrafts using several runways and gates avoiding collisions/conflicts.

Landing: an aircraft uses the runway, lands, and then taxis over to the terminal. Take-Off: an aircraft taxies to the runway and then takes off.

PROJECT - 5

AN AUTOMATED COMMUNITY PORTAL: Business in the 21st Century is above all BUSY. Distractions are everywhere. The current crop of “enterprise intranet portals” is often high noise and low value, despite the large capital expenditures it takes to stand them up. Email takes up 30 - 70% of an employee’s time. Chat and Instant Messaging are either in the enterprise or just around the corner. Meanwhile, management is tasked with unforeseen and unfunded leadership and change-agent roles as well as leadership development and succession management. What is needed is a simplified, repeatable process that enhances communications within an enterprise, while allowing management and peers to self-select future leaders and easily recognize high performance team members in a dynamic way. Additionally, the system should function as a general-purpose content management, business intelligence and peer-review application. Glass code's goal is to build that system. The software is released under a proprietary license, and will have the following features: Remote, unattended moderation of discussions However, it will have powerful discovery and business intelligence features, and be infinitely extendable, owing to a powerful API and adherence to Java platform standards. Encourages peer review and indicates for management potential leaders, strong team players and reinforces enterprise and team goals seamlessly and with zero administration.

PROJECT - 6

A CONTENT MANAGEMENT SYSTEM: The goal is to enable non-technical end users to easily publish, access, and share information over the web, while giving administrators and managers complete control over the presentation, style, security, and permissions.

FEATURES:

- Robust Permissions System
- Templates for easy custom site designs
- Total control over the content
- Search engine friendly URL's
- Role based publishing system
- Versioning control
- Visitor profiling

PROJECT - 7

AN AUCTION APPLICATION: Several commerce models exist and are the basis for a number of companies like eBay.com, pricellne.com etc. Design and implement an auction application that provides auctioning services. It should clearly model the various auctioneers, the bidding process, auctioning etc.

PROJECT - 8

A NOTES AND FILE MANAGEMENT SYSTEM: In the course of one's student years and professional career one produces a 1 lot of personal notes, documents. All these documents are usually kept 1 on papers or individual files on the computer. Either way the bulk of the information is often erased corrupted and eventually lost. The goal of this 1 project is to build a distributed software application that addresses this problem. The system will provide an interface to create, organize and manage personal notes through the Internet for multiple users. The system will also allow users to collaborate by assigning permissions for multiple users to view and edit notes.

PROJECT - 9

LIBRARY MANAGEMENT SYSTEM(LMS): The goal is to enable students and librarians to easily access and manage the library and run it smoothly. Each physical library item - book, tape cassette, CD, DVD, etc. could have its own item number. To support it, the items may be barcoded. The purpose of barcoding is to provide a unique and scannable identifier that links the barcoded physical item to the electronic record in the catalog. Barcode must be physically attached to the item, and barcode number is entered into the corresponding field in the electronic item record. Barcodes on library items could be replaced by RFID tags. The RFID tag can contain item's identifier, title, material type, etc. It is read by an RFID reader, without the need to open a book cover or CD/DVD case to scan it with barcode reader.

PROJECT - 10

HOSPITAL MANAGEMENT SYSTEM: Simulate to show and explain hospital structure, staff, and relationships with patients, and patient treatment terminology.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply software engineering principles, process models, and agile methodologies to plan and manage software development activities.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO2	Analyze software requirements and design models using UML diagrams to develop structured and maintainable software solutions.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO3	Evaluate software quality and reliability using metrics, testing strategies, and assurance standards to ensure conformance and maintainability.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO4	Create comprehensive software project documentation by integrating risk management, quality assurance, and industry-aligned design artifacts.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3

TEXT BOOKS:

1. Roger S. Pressman, “Software Engineering, A practitioner’s Approach”, 7th Edition, McGrawHill International edition, 2017.
2. Booch G., Rumbaugh J. and Jacobsons I, “The Unified Modeling Language User Guide”, 2nd Edition, Addison Wesley, 2010.

REFERENCE BOOKS:

1. Simon Sennet, Steve McRobb and Ray Farmer, “Object Oriented Systems Analysis and Design, 2nd Edition, 2014.
2. Deepak Jain “Software Engineering, Principles and Practices”, Oxford University Press,2010.
3. Pankaj Jalote, “Software Engineering, A Precise Approach”, Wiley India, 2010.

22TP302 QUANTITATIVE APTITUDE AND LOGICAL REASONING

Source: https://images.app.goo.gl/kvtVgA8TkvDCqLhj7

Hours Per Week :

L	T	P	C
1	2	-	2

PREREQUISITE KNOWLEDGE: Basic Logical Thinking and Problem Solving Ability.

COURSE DESCRIPTION AND OBJECTIVES:

The Students will be introduced to various Arithmetic and Reasoning Problems. The students will have acquaintance with various problems like Time & Work, Time & distance, Percentages, Profit & Loss etc. besides solving puzzles and Critical Reasoning.

MODULE-1

UNIT-1 **4L+8T+0P=12 Hours**

Number system, LCM & HCF of numbers, Percentage, Ratio and proportion, Profit, loss and discount, Average & Mixtures, Simple Interest & Compound interest.

UNIT-2 **4L+8T+0P=12 Hours**

Time and work, Time & distance, Problems on trains, Problems on ages, Permutation & Combinations, Probability.

PRACTICES:

- Each concept would be taught in detail in the class followed by 10 problems solved in the class. Students would have to solve 10 additional problems as homework assignment in each concept.

MODULE-2

UNIT-1 **4L+8T+0P=12 Hours**

Number series, Letter series, Analogy, Odd man out, Coding and decoding, Syllogisms- Statement & Conclusions, Puzzle test.

UNIT-2 **4L+8T+0P=12 Hours**

Blood relations, Direction sense test, Order & Ranking, Seating Arrangements, Calendar & Clocks.

PRACTICES:

- Each concept would be taught in detail in the class followed by 10 problems solved in the class. Students would have to solve 10 additional problems as home work assignment in each concept.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply mathematical concepts such as number systems, percentages, ratios, profit & loss, averages, interest, and time-based calculations to solve quantitative problems relevant to competitive and placement exams.	PO1, PO2, PO3, PO4, PO9, PO10, PO12, PSO1
CO2	Analyze and solve moderate to complex problems involving permutations, combinations, and probability using logical reasoning and appropriate mathematical tools..	PO1, PO2, PO3, PO4, PO9, PO10,PO12, PSO1
CO3	Evaluate reasoning patterns in series, analogies, syllogisms, and puzzles to enhance critical thinking and decision-making accuracy in competitive contexts.	PO1, PO2, PO3, PO4, PO9, PO10, PO12, PSO1
CO4	Solve real-world logical problems involving blood relations, directions, seating arrangements, calendars, and clocks using structured approaches and reasoning techniques.	PO1, PO2, PO3, PO4, PO9, PO10, PO12, PSO1

SKILLS:

- ✓ *Helps in developing and improving problem solving skills*
- ✓ *Allow students to develop critical thinking skills*

TEXT BOOKS:

1. R. S. Aggarwal- Quantitative Aptitude for Competitive Examinations- S. CHAND Publications- Revised Edition-2017.
2. ARIHANT- A New Approach To Verbal & Non-Verbal Reasoning- Arihant Publication- Revised Edition-2021.

REFERENCE BOOKS:

1. Trishna Knowledge Systems- Quantitative Aptitude for Competitive Examinations- Pearson Publication- First Edition- 2013.
2. R. S. Aggarwal- A Modern Approach to Verbal & Non-Verbal Reasoning-S. CHAND Publications- Revised Edition-2018.

22IT306 CLOUD COMPUTING

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Computer networks and Operating systems

COURSE DESCRIPTION AND OBJECTIVES:

This course is to provide the comprehensive and in-depth knowledge of Cloud Computing concepts, technologies, architecture and applications by introducing and researching state-of-the-art in Cloud Computing fundamental issues, technologies, applications and implementations. Another objective is to expose the students to frontier areas of Cloud Computing and information technology, while providing sufficient foundations to enable further study and research.

MODULE-1

UNIT-1 **12L+0T+8P=20 Hours**

CLOUD COMPUTING PARADIGMS:

Principles of Parallel and Distributed Computing: Parallel and Distributed Computing, Elements of Parallel and Distributed Computing, Technologies for distributed computing, Web2.0, Service Oriented Computing, Utility Oriented Computing.

Introduction: Cloud computing at a Glance, Historical Development, Building Cloud Computing Environments, Computing Platform and Technologies, Cloud reference model, Types of the Cloud. Cloud Platform in industry, AWS, GAE, MS Azure.

UNIT-2 **12L+0T+8P=20 Hours**

VIRTUAL MACHINES AND VIRTUALIZATION OF CLUSTERS AND DATA CENTERS:

Virtualization: Introduction, Characteristic of Virtualized Environments, Taxonomy of Virtualization Techniques, Virtualization and Cloud Computing, Pros and Cons of Virtualization.

Clusters and Data Centers: Implementation Levels of Virtualization, Virtualization Structures/Tools and Mechanisms, Virtualization of CPU, Memory, and I/O Devices, Virtual Clusters and Resource Management.

PRACTICES:

- Hardware architecture of Parallel processing.
- SISD, SIMD, MISD and MIMD basics.
- Components of Distributed systems.
- Distributed system Architecture.
- Client-Server and Peer-to-Peer architecture.
- Virtualized Environment.

Type II Hypervisor setup and configuration

- Create Virtual Machines (VM)
- Create Snapshot of all Virtual Machines
- Create clone of VM
- Configure Networking

Type I Hypervisor setup and configuration

- Virtual machines on Xen Migration of VMs
- Create a Virtual Network Switch
- Create Differencing Hard Disks for use by Virtual Machines
- Implementation of Desktop virtualization

SKILLS:

- ✓ *Understand the concept of cloud computing models.*
- ✓ *Development of cloud from the existing technologies.*
- ✓ *Knowledge on the various issues in cloud computing.*
- ✓ *Identify the types of virtualization.*
- ✓ *Understand the SOA benefits.*

- Implementation of Server virtualization – multiple servers on single system
- VM migration and Backup and recovery virtual machines in type I

MODULE-2

UNIT-1 **12L+0T+8P=20 Hours**

CLOUD ARCHITECTURE, SERVICES AND STORAGE:

Layered Cloud Architecture Design: NIST Cloud Computing Reference Architecture, Public, Private and Hybrid Clouds, laaS, PaaS, SaaS, Architectural Design Challenges, Cloud Storage, Storage-as-a-Service, Advantages of Cloud Storage, Cloud Storage Providers, S3.

UNIT-2 **12L+0T+8P=20 Hours**

SERVICE ORIENTED ARCHITECTURE:

Introduction: Services and Service-Oriented Architecture, Message-Oriented Middleware, Portals and Science Gateways, Discovery, Registries, Metadata, and Databases, Workflow in Service - Oriented Architectures.

PRACTICES:

- EC2 AWS – S3 bucket based static webpages.
- AWS – Local balancing and auto scaling.
- PaaS – GAE.
- Deployment of Open stack or Virtual box from the scratch.
- SaaS – Deployment of any SaaS application for an online collaborative too.
- Creating an application in Dropbox to store data securely.
- Develop a source code using Dropbox API for updating and retrieving files.
- Cloud application development using IBM Bluemix Cloud.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply foundational knowledge of cloud computing to launch and connect EC2 instances using various operating systems in AWS.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO2
CO2	Analyze the configuration and functionality of AWS storage services such as EBS, S3, Glacier, and EFS to determine suitable storage solutions.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO2
CO3	Evaluate the performance, scalability, and security of AWS-based network setups including public/private subnets, routing tables, and gateways in custom VPCs.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO2
CO4	Create secure and scalable cloud infrastructure by configuring AWS services such as EC2, S3, VPC, routing, and elastic IPs.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO2

TEXT BOOKS:

1. Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi, “Mastering Cloud Computing”, Tata Mcgraw Hill, 2013.
2. Kai Hwang, Geoffrey C. Fox, Jack G. Dongarra, “Distributed and Cloud Computing, From Parallel Processing to the Internet of Things”, Morgan Kaufmann Publishers, 2012.

REFERENCE BOOKS:

1. Rittinghouse, John W., and James F. Ransome, “Cloud Computing: Implementation, Management and Security”, CRC Press, 2017.
2. Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing - A Practical Approach”, Tata Mcgraw Hill, 2009.
3. George Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice)”, O’Reilly, 2009.

22IT307 MACHINE LEARNING

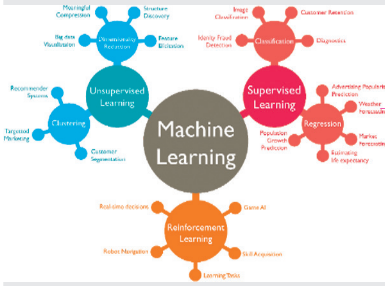


Image source: https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Basic statistics

COURSE DESCRIPTION AND OBJECTIVES:

The objective of this course is to familiarize the students with some basic learning algorithms and techniques and their applications, as well as general questions related to analyzing and handling large data sets. Several software libraries and data sets publicly available will be used to illustrate the application of these algorithms. The emphasis will be thus on machine learning algorithms and applications, with some broad explanation of the underlying principles.

MODULE-1

UNIT-1 **12L+0T+8P=20 Hours**

INTRODUCTION TO MACHINE LEARNING:

Machine Learning Fundamentals, Types of Machine Learning, Supervised, Unsupervised, Reinforcement, The Machine Learning process. Terminologies in ML. Testing ML algorithms: Overfitting, Training, Testing and Validation Sets Confusion matrix, Accuracy metrics, ROC Curve. Basic Statistics: Averages, Variance and Covariance, The Gaussian, The Bias, Variance trade off, Applications of Machine Learning.

UNIT-2 **12L+0T+8P = 20 Hours**

SUPERVISED LEARNING:

Regression: Linear Regression, Multivariate Regression. Classification: Linear Discriminant Analysis, Logistic Regression, K-Nearest Neighbor classifier. Decision Tree based methods for classification and Regression, Ensemble methods.

PRACTICES:

- Installation of Python Libraries/tools for Machine Learning.
- Data pre-processing using Python Machine Learning libraries.
- Design a model to predict the housing price from Boston Dataset using Multivariate Linear Regression.
- Build a classifier using Logistic Regression, k- Nearest Neighbor / Decision Tree to classify whether the given user will purchase a product or not from a social networking dataset.

MODULE-2

UNIT-1 **12L+0T+4P=16 Hours**

UNSUPERVISED LEARNING:

Clustering, K-Means clustering, Hierarchical clustering, The Curse of Dimensionality, Dimensionality Reduction, Principal Component Analysis, Probabilistic PCA, Independent Components analysis.

Perceptron, Multilayer perceptron, Back Propagation, Initialization, Training and Validation Support Vector Machines(SVM) as a linear and non-linear classifier, Limitations of SVM.

SKILLS:

- ✓ *Implement various classification algorithms.*
- ✓ *Implement variety of clustering techniques.*
- ✓ *Dealing with the data in high-dimensional space.*

UNIT-2 **12L+0T+12P=24 Hours**

ARTIFICIAL NEURAL NETWORKS:

Bayesian Networks, Learning Naive Bayes classifiers, Markov Models, Hidden Markov Models Sampling, Basic sampling methods, Monte Carlo, Reinforcement Learning.

PRACTICES:

- Segment a customer dataset based on the buying behaviour of customers using K-means/ Hierarchical clustering.
- Dimensionality reduction of any CSV/image dataset using Principal Component Analysis.
- Recognition of MNIST handwritten digits using Artificial Neural Network.
- Build an email spam classifier using SVM.
- Classify the given text segment as 'Positive' or 'Negative' statement using the Naïve Bayes Classifier.
- Predict future stock price of a company using Monte Carlo Simulation.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply machine learning fundamentals, model training techniques, validation strategies, and performance evaluation metrics across various algorithms such as regression, classification, clustering.	PO1, PO2, PO3, PO4, PO5, PO9, PO12, PSO1, PSO3
CO2	Analyze supervised and unsupervised learning algorithms including regression, decision trees, ensemble methods, k-means, PCA, to determine their strengths, limitations, and applicability for solving real-world problems	PO1, PO2, PO3, PO4, PO5, PO9, PO12, PSO1, PSO3
CO3	Evaluate the effectiveness of ML models using bias-variance trade-offs, dimensionality reduction, sampling methods, and statistical measures for performance optimization.	PO1, PO2, PO3, PO4, PO5, PO9, PO12, PSO1, PSO3
CO4	Create intelligent systems using a full ML pipeline by integrating concepts from supervised, unsupervised learning.	PO1, PO2, PO3, PO4, PO5, PO9, PO12, PSO1, PSO3

TEXT BOOKS:

1. Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, "Machine Learning",1st Edition Pearson Education, 2018.
2. Christopher Bishop, "Pattern Recognition and Machine Learning" 1st Edition, Springer, 2011.

REFERENCE BOOKS:

1. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", 1st Edition, MIT Press, 2012.
2. Stephen Marsland, "Machine Learning –An Algorithmic Perspective", 1st Edition, CRC Press, 2009.
3. Andreas C. Muller, "Introduction to Machine Learning with Python: A Guide for Data Scientists", 1st Edition, O'Reilly,2016.
4. Sebastian Raschka, "Python Machine Learning", 1st Edition, Packt Publishing, 2015.

IV

YEAR

B.Tech.

INFORMATION TECHNOLOGY

I SEMESTER

22IT401	-	Cryptography and Network Security
22IT402	-	Internet of Things
	-	Department Elective – 5
	-	Department Elective – 6
	-	Department Elective – 7
		Minor / Honors – 4

II SEMESTER

22IT403	-	Project Work
22IT404	-	Internship
	-	Minor / Honors – 5 (for Project)

COURSE CONTENTS

I SEM & II SEM

22IT401 CRYPTOGRAPHY AND NETWORK SECURITY

Image source: https://knowledge.tamu.qa/the-role-of-cryptography-in-our-information-based-society/

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Computer networks. Number theory and computational complexity

COURSE DESCRIPTION AND OBJECTIVES:

Cryptography refers to secure information and communication techniques derived from mathematical concepts and a set of rule-based calculations called algorithms, to transform messages in ways that are hard to decipher. To understand cryptography theories, algorithms and systems and understand necessary approaches and techniques to build protection mechanisms in order to secure computer networks.

MODULE-1

UNIT-1 **12L+0T+8P=20 Hours**

SECURITY CLASSICAL ENCRYPTION:

Security trends, Legal, Ethical and Professional Aspects of Security, Need for Security at Multiple levels, Security Policies, Model of network security, Security attacks, services and mechanisms, OSI security architecture.

Classical encryption techniques: substitution techniques, transposition techniques, steganography); Foundations of modern cryptography: perfect security, information theory, product cryptosystem, cryptanalysis.

UNIT-2 **12L+0T+8P=20 Hours**

SYMMETRIC KEY CRYPTOGRAPHY SYMMETRIC KEY CIPHERS:

Mathematics of Symmetric Key Cryptography: Algebraic structures, Modular arithmetic Euclid‟s algorithm, Congruence and matrices, Groups, Rings, Fields, Finite fields.

Symmetric Key Ciphers: SDES, Block cipher Principles of DES, Strength of DES, Differential and linear cryptanalysis, Block cipher design principles, Block cipher mode of operation, Evaluation criteria for AES, Advanced Encryption Standard, RC4, Key distribution.

PRACTICES:

- Understand the OSI Model.
- Understand Types of Network Devices.
- Know Network Defenses.
- Segregate Your Network.
- Place Your Security Devices Correctly.
- Use Network Address Translation.
- Don't Disable Personal Firewalls.
- Use Centralized Logging and Immediate Log Analysis.

SKILLS:

- ✓ *Problem-Solving Skills.*
- ✓ *Knowledge of Security across Various Platforms.*
- ✓ *Fundamental Computer Forensics Skills.*
- ✓ *An Understanding of Hacking.*

MODULE-2

UNIT-1 **12L+0T+8P=20 Hours**

KEY EXCHANGE AND DISTRIBUTION:

Mathematics of Asymmetric Key Cryptography: Primes, Primality Testing, Factorization, Euler‘s totient function, Fermat‘s and Euler‘s Theorem, Chinese Remainder Theorem, Exponentiation and logarithm.

Asymmetric Key Ciphers: RSA Cryptosystem, Key distribution, Key management, Diffie Hellman key exchange, ElGamal cryptosystem, Elliptic curve arithmetic, Elliptic curve cryptography. Authentication requirement, Authentication function, MAC, Hash function, Security of hash function and MAC, SHA, Digital signature and authentication protocols, DSS, Entity Authentication: Biometrics, Passwords, Challenge Response protocols, Authentication applications, Kerberos, X.509.

UNIT-2 **12L+0T+8P=20 Hours**

TYPES OF SYSTEM SECURITY:

Electronic Mail security, PGP, S/MIME, IP security, Web Security, SYSTEM SECURITY: Intruders, Malicious software, Viruses, Firewalls.

PRACTICES:

- Protect your data.
- Avoid pop-ups, unknown emails, and links.
- Use strong password protection and authentication.
- Connect to secure Wi-Fi.
- Enable firewall protection at work and at home.
- Invest in security systems.
- Install security software updates and back up your files.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply symmetric and asymmetric encryption techniques, hash functions, and digital signatures to protect data confidentiality, integrity, and authentication.	PO1, PO2, PO3, PO6, PO8, PO12, PSO1, PSO2
CO2	Analyze authentication protocols and cryptographic frameworks like Kerberos, IPsec, and PKI for secure network communication.	PO1, PO2, PO3, PO5, PO6, PO8, PO12, PSO1, PSO2
CO3	Evaluate the security services and mechanisms in place for web applications, email, and IP communication using SSL/TLS, S/MIME, and other protocols.	PO1, PO2, PO3, PO5, PO6, PO8, PO12, PSO1, PSO2
CO4	Evaluate the effectiveness of system-level protections including intrusion detection, firewalls, malware prevention, and password security mechanisms.	PO1, PO2, PO3, PO5, PO6, PO8, PO12, PSO1, PSO2

TEXT BOOKS:

1. William Stallings, “Cryptography and Network Security: Principles and Practice”, 3rd Edition, PHI, 2010.

REFERENCE BOOKS:

1. C K Shyamala, N Harini and Dr. T R Padmanabhan, “Cryptography and Network Security”, Wiley India Pvt.Ltd, 2012.
2. Behrouz A.Foruzan, ‘Cryptography and Network Security”, 1st Edition, Tata McGraw Hill, 2010.

22IT402 INTERNET OF THINGS

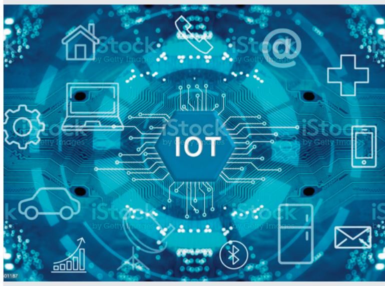


Image source: https://www.istockphoto.com/photo/internet-of-things-wireless-communication-network-abstract-image-visual-gm1184401187-333390319

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Basic knowledge of computer hardware and Computer programming

COURSE DESCRIPTION AND OBJECTIVES:

Students will be explored to the concepts and applications of Internet of Things, interconnection and integration of the physical world and the cyberspace. They are also able to design & develop IoT Devices and applications.

MODULE-1

UNIT-1 **12L+0T+4P=16Hours**

INTRODUCTION AND DOMAIN SPECIFIC IOT:

Introduction: Introduction to Internet of Things, Physical design of IoT, Logical design of IoT, IoT Enabling Technologies, IoT Levels.

Domain specific IoT: Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health & Life Style.

UNIT-2 **12L+0T+12P=24 Hours**

ELEMENTS OF IOT AND INTRODUCTION TO PYTHON MODULES AND PACKAGES:

Hardware Components-Computing (Arduino, Raspberry Pi), Sensors, Actuators, I/O interfaces, Communication Protocols (ZigBee, Bluetooth, 6LoPAN, and MQTT), Software Components-Programming API"s (using Python/Arduino).

Introduction: Functions, Modules, Packages, File handling, Date/ time Operations, Classes, Python package.

PRACTICES: (using Arduino Uno /Raspberry Pi board)

- Demonstration and study of Raspberry Pi board, GPIO Pins and familiarity of various sensors.
- Demonstration and study of other Hardware board of IoT such as Arduino Uno and NodeMCU Reading different types of data from the user and display that data using Scanner class.
- Design and Implementation of controlling LED.
- Design and Implementation of sensing light through LDR.

MODULE-2

UNIT-1 **12L+0T+4P=16 Hours**

M2M AND SYSTEM MANAGEMENT WITH NETCONF-YANG:

Introduction to M2M, Difference between IOT and M2M, SDN and NFV for IOT, Software defined Networking, Network function virtualization, Need for IOT systems management, Simple network management protocol, Limitations of SNMP, Network Operator Requirements, NETCONF, YANG.

UNIT-2 **12L+0T+12P=24 Hours**

IOT DESIGN METHODOLOGY:

Design methodology, Case study using weather monitoring; IOT Physical Devices & Endpoints, exemplary Device, Raspberry Pi interfaces, programming of IOT devices.

SKILLS:

- ✓ *Analyze the application areas of IOT.*
- ✓ *Writing Python scripts for IoT circuits.*
- ✓ *Design and develop IoT applications for given specific problem statement*

PRACTICES: (using Arduino Uno /Raspberry Pi board)

- Design and Implementation to find obstacles through sensor.
- Design and Implementation of detecting noise through microphone sensor.
- Design and Implementation of output devices through relay module.
- Design and Implementation of vibration sensor.
- Design and Implementation of sensing and display temperature.
- Design and Implementation of uploading sensor data into cloud using Python.
- Implement Python program to send email.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply IoT architecture, design principles, and enabling technologies to understand physical and logical components of IoT systems.	PO1, PO2,PO3, PO4, PO5, PO6, PO7, PO8, PO9,PO10,PO11, PO12, PSO1, PSO2, PSO3
CO2	Analyze domain-specific IoT applications and system management protocols including M2M, SDN, NFV, and NETCONF-YANG for real-time communication and control.	PO1, PO2,PO3, PO4, PO5, PO6, PO7, PO8, PO9,PO10,PO11, PO12, PSO1, PSO2, PSO3
CO3	Evaluate the use of Python programming and embedded platforms for efficient data acquisition, processing, and control in IoT environments.	PO1, PO2,PO3, PO4, PO5, PO6, PO7, PO8, PO9,PO10,PO11, PO12, PSO1, PSO2, PSO3
CO4	Create complete IoT-based solutions using sensors, devices like Raspberry Pi, and design methodologies to address real-world problems such as weather monitoring.	PO1, PO2,PO3, PO4, PO5, PO6, PO7, PO8, PO9,PO10,PO11, PO12, PSO1, PSO2, PSO3

TEXT BOOKS:

1. Arsh deep Bahga, Vijay Madisetti, “Internet of Things: A Hands-on-Approach”, 1st Edition, VPT, 2014.
2. Shriram K Vasudevan, Abhishek S Nagarajan, RMD Sundaram, “Internet of Things”, 1st Edition, John Wiley & Sons., 2019.

REFERENCE BOOKS:

1. Adrian McEwen, “Designing the Internet of Things”, 1st Edition, Wiley Publishers, 2013.
2. Daniel Kellmereit, “The Silent Intelligence: The Internet of Things”,1st Edition, DND Ventures LLC, 2013.

DEPT. ELECTIVES

INFORMATION TECHNOLOGY

B.Tech.

- 22IT827 - Open Source Web Technologies
- 22IT802 - Advanced Web Technologies
- 22IT823 - Mobile Computing
- 22IT822 - Mobile Application Development
- 22IT829 - Semantic Web and Social Networks
- 22IT818 - Full Stack Technologies
- 22IT808 - Computer Graphics
- 22IT815 - Digital Image Processing
- 22IT828 - Pattern Recognition
- 22IT819 - Image and Video Analytics
- 22IT810 - Computer Vision
- 22IT820 - Medical Image Analysis
- 22IT803 - Artificial Intelligence
- 22IT805 - Big Data Analytics
- 22IT813 - Data Science using Python
- 22IT804 - Artificial Neural Networks
- 22IT814 - Deep Learning
- 22IT824 - Natural Language Processing
- 22IT812 - Data Communications
- 22IT826 - Network Protocols
- 22IT831 - Wireless Networks
- 22IT811 - Cyber Security
- 22IT825 - Network Programming
- 22IT806 - Block Chain Technologies
- 22IT809 - Computer Organization and Architecture
- 22IT817 - Formal Languages and Automata Theory
- 22IT807 - Compiler Design
- 22IT801 - Advanced Computer Architecture
- 22IT816 - Distributed Systems
- 22IT821 - Microprocessors and Microcontrollers

COURSE CONTENTS

I SEM & II SEM

22IT827 OPEN SOURCE WEB TECHNOLOGIES

IMAGE SOURCE: https://thumbs.dreamstime.com/z/painted-hand-shows-concept-hologram-open-source-his-hand-painted-hand-shows-concept-hologram-open-source-his-hand-drawn-man-115398454.jpg

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: HTML, CSS and JavaScript

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the concepts pertaining to open-source technologies such as LINUX, MySQL, PHP, Apache web server, and various other tools used to develop web applications. In addition, this course is to offer insight into various open-source technologies to develop web applications.

MODULE-1

UNIT-1 **10L+0T+8P=18Hours**

OPEN SOURCE:

Open Source: Introduction, Open source operating System, Nature of open sources, Advantages, Application of open sources. Introduction to dynamic web content, Setting up a development server, Introduction to PHP, PHP functions and Objects, PHP arrays, Practical PHP.

UNIT-2 **14L+0T+8P=22 Hours**

OPEN SOURCE DATABASE:

Open Source Database: Introduction to MySQL, Accessing MySQL using PHP, querying a MySQL database with PHP, Practical MySQL, preventing hacking attempts, Using MySQL procedure.

Form Handling: Form Handling building forms, Retrieving submitted data, An example program, Cookies, Sessions, and authentication using cookies in PHP, HTTP authentication, Using sessions.

PRACTICES:

- Develop dynamic web content pages.
- Setting up a development server.
- Practice Basic PHP programs.
- Working with PHP functions and Objects.
- Create arrays using PHP.
- Create dynamic web pages using PHP.
- Practice basic MySQL queries.
- Access MySQL database using PHP.
- Create Database Connectivity with PHP & MySQL.
- Working with MySQL procedures.
- Create forms using PHP and retrieve the data from the database.
- Work with Cookies and Sessions in PHP.

MODULE-2

UNIT-1 **10L+0T+8P=18 Hours**

HTML5:

Introduction: to HTML5, The HTML5 Canvas, HTML5 audio, and video, Bringing it all together by designing a social networking site.

SKILLS:

- ✓ *Use the Open source technologies.*
- ✓ *Develop dynamic web pages.*
- ✓ *Design a social networking website.*
- ✓ *Apply the AngularJS concepts to dynamic websites.*
- ✓ *Create animations using AngularJS.*

UNIT-2 **14L+0T+8P=22 Hours**

ANGULAR JS:

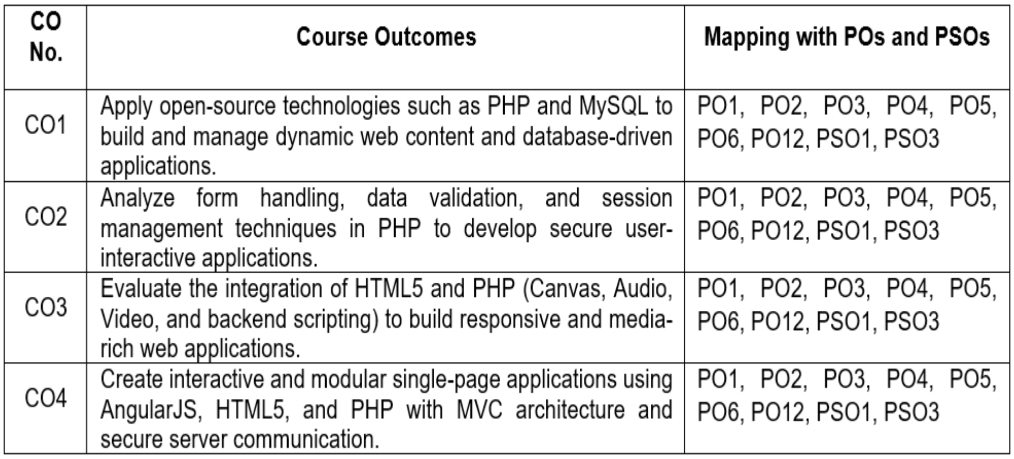
Angular JS: The basics of AngularJS, Introduction MVC, Filters, and modules, Directives, Working with Forms, Services and server communication, Organizing views, Angular JS animation.

PRACTICES:

- Practice basic HTML5 tags.
- Usage of HTML5 audio, video, and canvas tags.
- Build a Social Networking Website.
- Practice basic AngularJS.
- Working with forms, services, and server communication through AngularJS.
- Using views in AngularJS applications.
- Create animations using AngularJS.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply open-source technologies such as PHP and MySQL to build and manage dynamic web content and database-driven applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3
CO2	Analyze form handling, data validation, and session management techniques in PHP to develop secure user-interactive applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3
CO3	Evaluate the integration of HTML5 and PHP (Canvas, Audio, Video, and backend scripting) to build responsive and media-rich web applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3
CO4	Create interactive and modular single-page applications using AngularJS, HTML5, and PHP with MVC architecture and secure server communication.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3

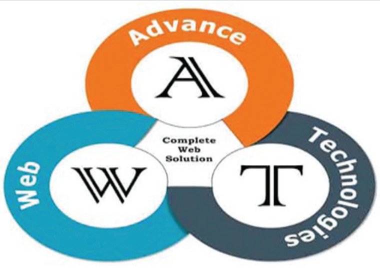
TEXT BOOKS:

1. Godfrey Boyle, “Renewable Energy, Power for a Sustainable Future”, 1st edition, Oxford URobin Nixon, “Learning PHP, MySQL & JavaScript WITH JQUERY, CSS & HTML5”, 5th Edition, O’Reilly, 2018.
2. Andrew Grant, “Beginning Angular JS”, 1st Edition, Apress, 2014.

REFERENCE BOOKS:

1. Steve Prettyman, “Learn PHP 7 Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL”, 1st Edition, Apress, 2015.
2. Adrian W. West and Steve Prettyman, “Practical PHP 7, MySQL 8, and MariaDB Website Databases: A Simplified Approach to Developing Database-Driven Websites”, 2nd Edition, A Press, 2018.

22IT802 ADVANCED WEB TECHNOLOGIES



Source: https://content.jdmagicbox.com/comp/allahabad/l5/0532px532.x532.170901125535.w3l5/catalogue/advance-web-technologies-it-software-solution-kareli-allahabad-it-solution-providers-zmftm6y0ku.jpg?clr=293d3d

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Computer programming and Web document languages

COURSE DESCRIPTION AND OBJECTIVES:

This course is intended for students who are interested in designing their own websites. Over the course, students will learn how to build their own websites using HTML and CSS. They will learn different techniques intended to help them customize, organize, and style their web pages. JavaScript is a loosely-typed client-side scripting language that executes in the user's web browser. A web page without JavaScript is unimaginable today. By the end of the course, students will have a strong understanding and foundation for web development.

MODULE-1

UNIT-1 **16L+0T +6P=22 Hours**

INTRODUCTION TO WEB TECHOLOGIES GENERIC VIEW OF WEBSITES:

Introduction to Web Technology: Hypertext Mark-up Language and its components, HTML tags and attributes, Text formatting tags, List tags, Image tags, HTML tables, HTML Forms, Document Object Model (DOM), Cascading Style Sheets – Inline Style, Embedded Style, External Style Sheet, Imported Style Sheet, Ruleset, @ rule, Contextual Selector, Attribute Selector, CSS Properties, JavaScript - Data types, Operators, Variables, length, substring, Conditional Statements - if, Loops - for, & Functions, HTML DOM and JavaScript - Finding HTML Elements, Changing HTML elements, DOM events).

UNIT-2 **8L+0T+10P=18 Hours**

EVENTS HANDLERS & FORMS IN JAVA SCRIPT - MESSAGING & TIMING EVENTS IN JAVA SCRIPT:

Events Handlers & Forms in Java Script: Define Events, Events in JavaScripts, Event Handlers, this keyword, Event handlers in JavaScripts, Emulating Events in java scripting, onLoad and onUnload Event Handlers, Web-Hopping with window.open(), Resetting Event Handlers.

Messaging & Timing Events in Java Script: Alert Box: syntax & its example, Confirm Box: syntax & its example, Prompt Box: syntax & its example, Line Breaks: syntax & its example, JavaScript Timing Events, setInterval() Method, window.clearInterval() method, setTimeout() Method, window.clearTimeout() method.

PRACTICES:

- Create a HTML page having four frames named a) Top b) Center c) Bottom d) Left The top frame should contain company logo and title. The bottom frame should contain copy right information. The left frame should contain various links like Home, Products, Services, Branches, About us, etc. When clicked on those links, the contents should appear in the display on to center frame.
- Design a catalogue page that should contain the details of all the books available in the web site in a table. The details should contain the following: a) Snap shot of Cover Page b) Author Name c) Publisher. d) Price. e) Add to cart button.
- Create a Time-table schedule of your current semester using HTML table.
- Create a HTML document to demonstrate Form Elements that includes Form, input-text, password, radio, checkbox, hidden, button, submit, reset, label, text area, select, option, file upload.

- Write a HTML program with at least two <h1>, two images, two buttons and appropriate CSS to display, a) All <h1> with font-size 12pt, and bold in Verdana font using Inline CSS. b) All <img> with border color yellow, thickness 10px using Document Level CSS c) All <input type='button'> should change background color to red on mouse over them using External CSS.
- Design a HTML page having a text box and four buttons viz Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate java script function should be called to display the following: a) factorial of that number b) fibonacci series up to that number c) prime numbers up to that number d) is it palindrome or not?
- Validate the Registration, user login and payment by credit card pages using JavaScript.
- Write a JavaScript program to convert temperatures to and from celsius, Fahrenheit.
- Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
- Write a JavaScript code that displays text "TEXT-GROWING" with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays "TEXT-SHRINKING" in BLUE color. Then the font size decreases to 5pt.

MODULE-2

UNIT-1 **12L+0T+6P=18 Hours**

ANGULAR JS:

Angular JS: Introduction to Angular JS, Basics and Syntax of Angular JS, Features, Advantages, Application Structure, Basics of routes and navigation, MVC with Angular JS, Services. Modules, Directives, Routes, Angular JS Forms and Validations, Data binding.

UNIT-2 **12L+0T+10P=22 Hours**

NODE JS:

Node JS: Introduction to Node JS, Setup Node JS Environment, Package Manager, Features, Console Object, REPL Terminal, Concept of Callbacks. Events and Event Loop, timers, Error Handling, Buffers, Streams, Node.js Webserver – Server and Clients, Creating a simple server, Rendering HTML, Rendering JSON Data, Configuring Routes, Working with Express JS,Debugging, Event- Emitter class, Inheriting Events, Returning event emitter..

PRACTICES:

- Implement the following client side validations using AngularJS.
- Make sure the username starts with an uppercase letter b) The username must have at least one digit included c) Ensure that Email is valid d) Ensure the password length is between 8 to 20 characters e) Make sure the password contains at least one uppercase letter.
- Form Validation using Angular js.
- Creating single page website using Angular JS.
- Build a simple Web application that returns a message as "Hello World" when user request the server using Node JS.
- Write a program to send JSON responses in Node JS.
- Write a program to send HTML as a response in Node JS.
- Create different routes to support multiple responses in Node JS.
- Create Express routers to serve different HTTP methods such as GET, POST, PUT, DELETE HEAD.

SKILLS:

- ✓ *Define the fundamental ideas and standards underlying Web Service Technology.*
- ✓ *Discuss concepts at the frontier of industrial practice and emerging standards.*
- ✓ *Differentiate the major frameworks allowing to develop web services and cloud applications and assess their suitability for specific usage scenarios.*
- ✓ *Understand the link between the concepts of services and business processes and discuss and critique related standards.*
- ✓ *Develop business processes using the Workflow foundation.*
- ✓ *Develop and deploy web services using appropriate Microsoft technologies.*

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply fundamental concepts of HTML, CSS, and JavaScript to design and style interactive webpages.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3
CO2	Analyze JavaScript events, forms, and timing mechanisms to build responsive web interfaces and manage user interactions effectively.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3
CO3	Evaluate the use of AngularJS for dynamic, component-based development and data-driven single-page applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3
CO4	Create scalable and event-driven server-side applications using NodeJS with Express, routing, and real-time data handling.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3

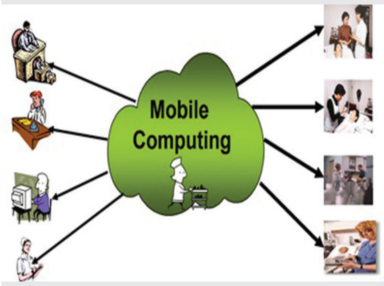
TEXT BOOKS:

1. Jon Duckett, “Beginning Web Programming with HTML, XHTML, and CSS”, 2nd Edition, Wiley India Pvt. Ltd, 2008.
2. Brad Dayley, Brendan Dayley, and Caleb Dayley, “Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications”, 2nd Edition, Pearson Education, 2018.

REFERENCE BOOKS:

1. Paul Deitel, Harvey Deitel, Abbey Deitel, “Internet & World Wide Web How to Program”, 5th Edition, Pearson Education, 2012.
2. Manuel Kiessling, “The Node Craftsman Book”, Packt Publisher, 2017. 4. Marty Hall and Larry Brown, “Core Servlets and Java Server pages Vol. 1: Core Technologies”, 2nd Edition, Pearson, 2004.

22IT823 MOBILE COMPUTING



Source: https://noidatut.com/mobile-computing/mobile-computing.php

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Basic Communication

COURSE DESCRIPTION AND OBJECTIVES:

This course helps a student to design effective mobile applications using the Android development environment. The main objective of this course is to create user-friendly applications that involve design of layout, windows components, and multiple screens with one-touch options. Also allows student to students to understand the of wireless and mobile communication technologies and applications.

MODULE-1

UNIT-1 6L+6T+0P=12 Hours

INTRODUCTION:

Introduction to Mobile Computing, Applications of Mobile Computing, Generations of Mobile Communication Technologies, Multiplexing, Spread spectrum, MAC Protocols, SDMA, TDMA, FDMA, CDMA.

UNIT-2 10L+10T+0P=20 Hours

MOBILE TELECOMMUNICATION SYSTEM:

Introduction to Cellular Systems, GSM Services & Architecture, Protocols, Connection Establishment, Frequency Allocation, Routing, Mobility Management, Security, GPRS, UMTS Architecture, Handover, Security.

PRACTICES:

- Develop and demonstrate mobile applications using various tools
- Articulate the knowledge of GSM, CDMA & Bluetooth technologies and demonstrate it.
- Simulation of frequency reuse, hidden/exposed terminal problem
- Implement security algorithms for mobile communication network
- Managing screen orientation

MODULE-2

UNIT-1 8L+8T+0P=16 Hours

MOBILE NETWORK, TRANSPORT AND APPLICATION LAYER:

Mobile IP, DHCP, AdHoc, Proactive protocol, DSDV, Reactive Routing Protocols: DSR, AODV, Hybrid routing, ZRP, Multicast Routing, ODMRP, Vehicular Ad Hoc networks (VANET), MANET Vs VANET, Security. Mobile TCP, WAP Architecture: WDP, WTLS, WTP, WSP, WAE, WTA Architecture, WML

UNIT-2 8L+8T+0P=16 Hours

MOBILE PLATFORMS AND APPLICATIONS:

Mobile Device Operating Systems, Special Constraints & Requirements, Commercial Mobile Operating Systems, Software Development Kit: iOS, Android, BlackBerry, Windows Phone, M-Commerce, Structure Pros & Cons, Mobile Payment System, Security Issues

PRACTICES:

- The cellular frequency reuse concept to find the co-channel cells for a particular cell. Design a game-based application on the above concept.
- Implementation a Bluetooth network with application as transfer of a file from one device to another.
- To implement a basic function of Code Division Multiple Access(CDMA) to test the orthogonality and autocorrelation of a code to be used for CDMA operation.
- Basic views.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the concepts of mobile communication technologies such as GSM, CDMA, and multiplexing techniques for enabling wireless connectivity.	PO1, PO2, PO3, PO5, PO6, PO12, PSO1, PSO3
CO2	Analyze the protocols and architectures of mobile telecommunication systems like GSM, GPRS, and UMTS for routing, handover, and security management.	PO1, PO2, PO3, PO5, PO6, PO12, PSO1, PSO3
CO3	Evaluate mobile network protocols including Mobile IP, AdHoc, AODV, VANET, and their performance in various mobility scenarios.	PO1, PO2, PO3, PO5, PO6, PO12, PSO1, PSO3
CO4	Create secure and scalable mobile applications using modern mobile platforms (iOS, Android, Windows) with attention to m-commerce and device constraints.	PO1, PO2, PO3, PO5, PO6, PO12, PSO1, PSO3

TEXT BOOKS:

1. Jochen Schiller, “Mobile Communications”, 2nd Edition, PHI, 2013.
2. Prasant Kumar Pattnaik, Rajib Mall, “Fundamentals of Mobile Computing”, 1st Edition, PHI Learning Pvt. Ltd, 2012.

REFERENCE BOOKS:

1. Dharma Prakash Agarval, Qing and An Zeng, “Introduction to Wireless and Mobile systems”, 1st Edition, Thomson Asia Pvt Ltd, 2005.
2. Uwe Hansmann, Lothar Merk, Martin S. Nicklons and Thomas Stober, “Principles of Mobile Computing”, Springer, 2003.
3. William.C.Y.Lee, “Mobile Cellular Telecommunications-Analog and Digital Systems”, 2nd Edition, TataMcGraw Hill, 2006.

SKILLS:

- ✓ *Implement Mobile node discovery. 33*
- ✓ *Implementation of GSM security algorithms (A3/A5/A8)*
- ✓ *Illustration of Hidden Terminal Problem (NS-2).*
- ✓ *Develop an application that uses GUI components.*

Image source: https://5.imimg.com/data5/FH/UC/MY-9120378/android-app-development-company-500x500.png

22IT822 MOBILE APPLICATION DEVELOPMENT

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Basics of Object Oriented Programming through JAVA, XML.

COURSE DESCRIPTION AND OBJECTIVES:

This course helps a student to design effective mobile applications using the Android development environment. The main objective of this course is to create user-friendly applications that involve design of layout, window components, and multiple screens with one- touch options.

MODULE-1

UNIT-1 **8L+6T+0P=14 Hours**

GETTING STARTED WITH ANDROID PROGRAMMING AND ACTIVITIES:

Introduction to Android: Android introduction, Versions of android, Features of android, Architecture, Devices in the market, Developer community.

Understanding Activities: Life cycle of an activity, applying styles and themes, Types of dialog boxes, Types of Intents, Resolving Intent filter collision, Returning Results from an Intent, passing the data using Intent Object.

UNIT-2 **8L+10T+0P=18 Hours**

INTENTS AND FRAGMENTS:

Intents: understanding about Intent Objet and Intent filters, calling built in apps using Intents.

Fragments: Fragment types, Life Cycle, Interaction between Fragments.

PRACTICES:

- Installation of Android studio, its required tools and AVD.
- Displaying the hello world message in AVD.
- Creating a basic activity and applying themes, styles to it.
- Displaying various types of Dialog objects.
- Linking activities with Intents.
- Passing data using intent object.
- Usage of Fragments and adding them dynamically to the application.
- Communication between fragments.

MODULE-2

UNIT-1 **8L+6T+0P=14 Hours**

ANDROID UI DESIGN & DISPLAY ORIENTATION:

Android UI Design: Understand components of a screen, View, View Groups, Linear Layout, Absolute Layout, Table Layout, Relative Layout, Frame Layout, Scroll View.

Display Orientation: Anchoring Views, Resizing and Repositioning Views, managing changes to screen orientation, utilizing the action bar, Creating UI programmatically.

UNIT-2 **8L+10T+0P=18 Hours**

UNDERSTANDING BASIC VIEWS:

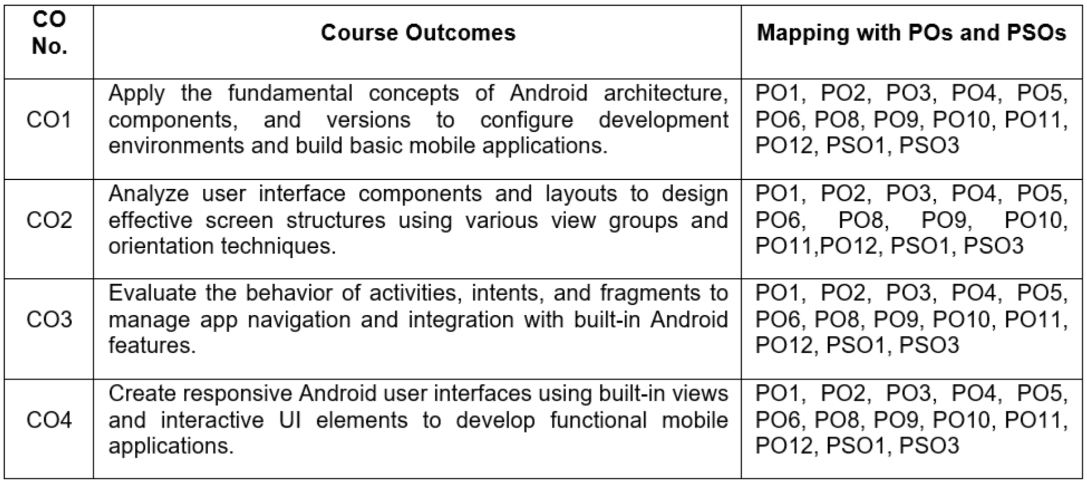
Basic Views: Designing UI with basic views, Progress bar view, Auto complete Text View.

PRACTICES:

- Design an application with various Layouts.
- Designing Action bar to the application.
- Handling the basic view events of the application.
- Design an application to implement AutoComplete Text View.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the fundamental concepts of Android architecture, components, and versions to configure development environments and build basic mobile applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO2	Analyze user interface components and layouts to design effective screen structures using various view groups and orientation techniques.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11,PO12, PSO1, PSO3
CO3	Evaluate the behavior of activities, intents, and fragments to manage app navigation and integration with built-in Android features.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO4	Create responsive Android user interfaces using built-in views and interactive UI elements to develop functional mobile applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3

TEXT BOOK:

1. Wei-Meng Lee, “Beginning Android 4 Application Development”, 1st Edition, John Wiley & Sons, 2012.

REFERENCE BOOKS:

1. RaimonRefolsMontane, Laurence Dawson, “Learning and Android Application Development”, 1st Edition, PACKT Publishing, 2016.
2. Reto Meier, “Professional Android 4 Application Development”, 3rd Edition, Wrox, 2012.
3. Adam Gerber and Clifton Craig, “Learn Android Studio”, 1st Edition, Apress, 2015.

SKILLS:

✓ *Understanding mobile applications for user requirements.*

✓ *Usage of various components of Android operating system.*

✓ *Utilization of activities, intents, layouts and views for content.*

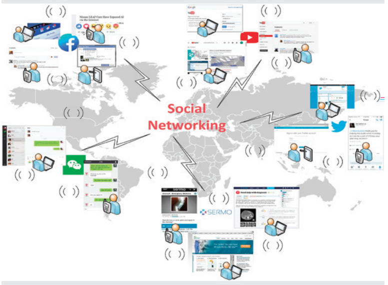


Image source: https://onlinelibrary.wiley.com/doi/full/10.1002/dac.4424

22IT829 SEMANTIC WEB AND SOCIAL NETWORKS

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Web Technology and Web Service

COURSE DESCRIPTION AND OBJECTIVES:

The objective of this course is to provide the comprehensive and in-depth knowledge of Semantic Web and Social Networks concepts, technologies, architectures and applications by introducing and researching state-of-the-art in Semantic Web and Social Networks fundamental technologies, applications and implementations.

MODULE-1

UNIT-1 **10L+6T+0P=16 Hours**

SEMANTIC WEB:

Introduction: Defining the Semantic Web, Comparison of WWW and SW, comparison of Knowledgebase and Relational databases, Major semantic web components, expressing semantic data, sharing data, making data dynamic and flexible, semantic web roadblock, semantic web myths, semantic web hype, Understanding Semantic Web Origins, Exploring Semantic Web Examples.

UNIT-2 **6L+10T+0P=16 Hours**

SEMANTIC WEB DEVELOPMENT AND ONTOLOGY:

Setting up Semantic Web Development Environment: Programming the Hello Semantic Web World Application, Modeling information, Semantic web information Model, RDF-Semantic on web, Expressing semantics in RDF.

Ontology: An ontology language for the web, Introduction to Ontologies, Overview of ontology elements, Elements of an ontology. Ontology based knowledge representation: Resource Description Framework(RDF) – Web ontology language (OWL).

PRACTICES:

- Semantic Web Vs WWW
- Components of semantic web
- Connecting web data and representing information.
- Distinguish between knowledge base and relational database
- Convert dumb data to smart linked data
- Semantic web technical standards RDF and OWL
- Ontology based knowledge representation
- Case study on BBC-Media management

MODULE-2

UNIT-1 **10L+6T+0P=16 Hours**

SOCIAL NETWORK ANALYSIS FUNDAMENTALS:

Introduction: Statistical Properties of Social Networks, Social Network Analysis, Development of Social Network analysis, Key concept of social network analysis, Semantic Technology for Social Network Analysis, Electronic discussion network, Blog and online communities, Web based networks.

UNIT-2 **6L+10T+0P=16 Hours**

MODELING AND VISUALIZATION:

Visualizing Online Social Networks: A Taxonomy of Visualizations, Graph Representation, Centrality, Clustering, Node-Edge Diagrams, Visualizing Social Networks with Matrix Based Representations, Node-Link Diagrams, Hybrid Representations, Modelling and aggregating social network data, Random Walks and their Applications, Use of Hadoop and MapReduce, Ontological representation of social individuals and relationships.

PRACTICES:

- Differences about social network data.
- Demonstrate proficiency the use of social network analysis and social network developer tools.
- Demonstrate components of the social network.
- Model and visualize the social network.
- Analyze the users in the social network.
- Comprehend the evolution of the social network.
- Build a social network data set from existing social networking sites.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the concepts of Semantic Web and its architecture to represent and share data meaningfully using RDF and OWL.	PO1, PO2, PO3, PO5, PSO1
CO2	Analyze ontology elements, knowledge models, and semantic relationships for building intelligent web systems.	PO1, PO2, PO4, PO5, PO6, PSO1, PSO2
CO3	Evaluate statistical and structural properties of social networks using SNA principles and explore social interactions with semantic technologies.	PO1, PO2, PO3, PO4, PO6, PO10, PSO1, PSO2
CO4	Create visualizations and models of online social networks using graph techniques, matrix representations, and big data tools like Hadoop and MapReduce.	PO1, PO3, PO5, PO11, PO12, PSO1, PSO3

TEXT BOOKS:

1. Peter Mika, "Social Networks and the Semantic Web", Springer, 1st Edition, 2010.
2. Charu C. Aggarwal, "Social Network Data Analytics", Springer; 2011

REFERENCE BOOKS:

1. Ajith Abraham, Aboul Ella Hassanien, Václav Snášel, "Computational Social Network Analysis: Trends, Tools and Research Advances", Springer, 2012.
2. John Hebeler, Matthew Fisher, Ryan Blace, Andrew Perez-Lopez, "Semantic Web Programming", 1st Edition, Wiley Press, 2009.
3. Borko Furht, "Handbook of Social Network Technologies and Applications", Springer, 1st Edition, 2011.

SKILLS:

- ✓ *Understanding Various Social Networking Sites.*
- ✓ *Introduction to Social Network Analysis.*
- ✓ *Understand the components of the social network.*
- ✓ *Develop model and visualize the social network.*

Image source: https://dataflog.com/wp-content/uploads/2022/02/2590506-750x420.jpg

22IT818 FULL STACK TECHNOLOGIES

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Python, DBMS, Web Technologies, Web Services, Android

COURSE DESCRIPTION AND OBJECTIVES:

This course explores and demonstrate Understand the advantages of front end and back end tech. The objective of this course is to offer the knowledge and skill on the basic principles and concepts in complete web development.

MODULE-1

UNIT-1 **12L+0T+8P=20 Hours**

INTRODUCTION TO WEB PROGRAMMING:

Git - An Introduction to Version Control, Git, Command-line Scripting, Basic HTML, CSS

Bootstrap - Overview, Environment setup, Precompiled Bootstrap, Source Code, Grid System, Bootstrap CSS Overview, Typography, Code, Tables, Forms, Helper Classes, Responsive Utilities, Glyph icons, Dropdowns, Navigation Elements, Breadcrumb, Pagination, Badges, Progress bars, Plugins Overview, Transition Plugin, Model Plugin, Dropdown Plugin, Scroll spy Plugin, Tab Plugin, Tooltip Plugin, Popover Plugin, Alert Plugin, Button Plugin, Collapse Plugin, Carousel Plugin, Affix Plugin.

UNIT-2 **12L+0T+8P=20 Hours**

CLOUD AND DEVOPS:

Motivation-Cloud as a Platform, Operations, Deployment Pipeline: Over all Architecture Building and Testing, Deployment, Case study: Migrating to Micro services.

Origin of DevOps - The developers versus operations dilemma, Key characteristics of a DevOps culture, deploying a Web Application, Creating and configuring an account, Creating a web server, Managing infrastructure with Cloud Formation, Adding a configuration management system.

PRACTICES:

- Configure the web application in Bootstrap.
- Development of web application using DevOps.
- Configure the web server for web application using DevOps.

MODULE-2

UNIT-1 **12L+0T+8P=20 Hours**

DJANGO AND MONGODB:

Django - Creating the Project, Running the Development Server, Creating the Application, designing a Model, setting up the Database, Setting up the Application, Dynamic Web Sites, Communication, Data Storage, Presentation.

Introduction to Mongodb - JSON and MongoDB, adopting a Non, relational Approach, opting for Performance vs. Features Running the Database Anywhere, Generating or Creating a Key, Using Keys and Values, Implementing Collections.

UNIT-2 **12L+0T+8P=20 Hours**

ANGULAR JS, NODE JS AND EXPRESS FRAMEWORK:

Introduction to Angular JS - Introduction, Features, Angular JS Model-View, Controller, Expression, Directives and Controllers, Angular JS Modules, Arrays, Working with ng-model, Working with Forms, Form Validation, Error Handling with Forms, Nested Forms with ng-form, Other Form Controls.

Node JS and Express Framework - Introduction, Using the Terminals, Editors, Building a Webserver with Node, The HTTP Module, Views and Layouts, Middleware, Routing, Form Handling with Express, The Request and Response Objects, Handle bars, Comments and Blocks, Polymorphism.

PRACTICES:

- Battleships Multiplayer Gaming Application.
- Angular 2 chess game.
- Build a web application using Bootstrap.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply front-end design concepts using HTML, CSS, Bootstrap, and Git version control to build responsive web interfaces.	PO1, PO2, PO3, PO4, PO5, PO6, PO9, PO10, PO11, PO12, PSO1, PSO3
CO2	Analyze cloud infrastructure, deployment pipelines, and DevOps culture to streamline software delivery and operations.	PO1, PO2, PO3, PO4, PO5, PO6, PO9, PO10, PO11, PO12, PSO1, PSO3
CO3	Evaluate dynamic backend web development using Django and data management using MongoDB for designing scalable and flexible systems.	PO1, PO2, PO3, PO4, PO5, PO6, PO9, PO10, PO11, PO12, PSO1, PSO3
CO4	Create full-stack applications using AngularJS for front-end and Node.js with Express.js for backend including routing, form handling, and middleware integration.	PO1, PO2, PO3, PO4, PO5, PO6, PO9, PO10, PO11, PO12, PSO1, PSO3

TEXT BOOKS:

1. Len Bass, Ingo Weber and Liming Zhu, “DevOps: A Software Architect’s Perspective”, Pearson Education, 2016
2. Adam Freeman - ProAngular JS, 1st Edition, Apress, 2014

REFERENCE BOOKS:

1. Agus Kurniawan, “AngularJS Programming by Example”, 1st Edition, PE Press, 2014.
2. Kyle Banker, Peter Bakkum, Shaun Verch, Dough Garrett, Tim Hawkins, “MongoDB in Action”, 2nd Edition, Manning Publications, 2016.
3. Steve Hoberman, “Data Modeling for MongoDB”, 1st Edition, Technics Publication, 2014.
4. Shyam Seshadri, Brad Green, “AngularJS: Up and Running: Enhanced Productivity with Structured Web Apps”, Apress, 2015.
5. Evan M. Hahn,“Express in Action”, 1st Edition, Manning Publications, 2014.

SKILLS:

- ✓ *Able to apply various fundamentals for problem solving using python.*
- ✓ *Able to implement DevOps.*
- ✓ *Able to create object-oriented solution by applying various concept like polymorphism, inheritance and package with python programming.*

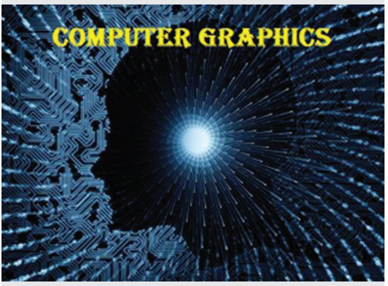


Image source: https://dataflaq.com/wp-content/uploads/2022/02/2590506-750x420.jpg

22IT808 COMPUTER GRAPHICS

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Engineering mathematics and Programming

COURSE DESCRIPTION AND OBJECTIVES:

This course will explore core topics of computer graphics such as the GPU pipeline, essential mathematics for computer graphics, viewing 3D objects, common geometric data structures, materials modelling, basic drawing, image processing, the human visual system, color science, illumination and rendering.

MODULE-1

UNIT-1 **8L+6T+0P=14 Hours**

COMPUTER GRAPHICS AND GPU PROGRAMMING:

Introduction: Computer graphics and purpose of course, Brief history of computer graphics, I/O graphics, Overview of the course.

Graphics and GPU Programming: GPU programming theory and practice. Pipelines - Graphics programming + Event driven pipelines. OpenGL and DirectX APIs - key differences. Key CPU vs. GPU programming differences. Shaders. Desktop, Mobile and Web graphics differences.

UNIT-2 **8L+10T+0P=18 Hours**

MATHEMATICS FOR COMPUTER GRAPHICS AND GEOMETRY:

Essential Mathematics for Computer Graphics: The Virtual Camera. Euclidean space and basic terminology. Rigid body Transformations. Transformations: Translation, Scale, Rotation. 2D vs 3D transformations.

Geometry: Primitive Objects. Constructive solid geometry. Polygons. Voxels. Boundary representation. Level of Detail and Tessellation. Acceleration Data structures.

PRACTICES:

- Demonstrate GPU.
- Practice in openGL.
- DirectX APIs.
- Distinguishing CPU and GPU.
- Mobile and web graphics differences.
- Calculate Euclidean space.
- Rigid body transformation.
- 2D and 3D scalling.

MODULE-2

UNIT-1 **8L+6T+0P=14 Hours**

MATERIALS AND BASIC DRAWING:

Materials: Texture Mapping. Bump Mapping. Capturing the surface colour of objects – challenges. Light to Surface Interactions. OpenGL lighting/reflectance model. Bidirectional Reflectance Distribution Functions. Bidirectional Subsurface Scattering Reflection Distribution Functions. Bidirectional Texture Functions.

Basic Drawing Fonts Basics. Drawing lines. Half ray test. Scan conversion. Rasterisation.

UNIT-2 **8L+10T+0P=18 Hours**

HUMAN FACTORS ON COLOUR SCIENCE AND ILLUMINATION:

Human Factors and Basic Colour Science: Human visual system: basics of how we perceive the world. Strengths and weaknesses of the human visual system. Graphics and usability – best practices for designing GUIs and graphics applications. Human eye anatomy. Measuring Light: Radiometry vs. Photometry. Colour models: RGB, CMYK, HVS, XYZ, Spectral Radiance.

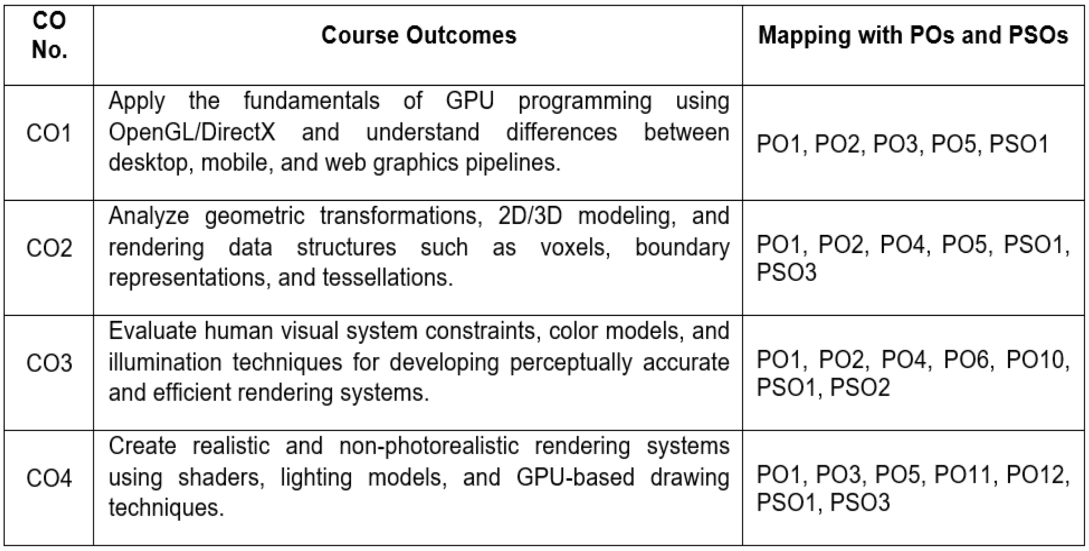
Illumination and Rendering: Defining Realism. Image Synthesis Validation. The challenges in computing light. Optics Models. Real-time vs. offline rendering. Global Illumination vs. Local Illumination. Types of Light Sources. Shadows. Non-photorealistic Rendering.

PRACTICES:

- Best practices for designing GUIs.
- Measuring light.
- Colour models.
- Image synthesis validation.
- Optics models.
- Types of light sources.
- Shadows.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the fundamentals of GPU programming using OpenGL/DirectX and understand differences between desktop, mobile, and web graphics pipelines.	PO1, PO2, PO3, PO5, PSO1
CO2	Analyze geometric transformations, 2D/3D modeling, and rendering data structures such as voxels, boundary representations, and tessellations.	PO1, PO2, PO4, PO5, PSO1, PSO3
CO3	Evaluate human visual system constraints, color models, and illumination techniques for developing perceptually accurate and efficient rendering systems.	PO1, PO2, PO4, PO6, PO10, PSO1, PSO2
CO4	Create realistic and non-photorealistic rendering systems using shaders, lighting models, and GPU-based drawing techniques.	PO1, PO3, PO5, PO11, PO12, PSO1, PSO3

TEXT BOOK:

1. Peter Shirley, “Fundamentals of Computer Graphics”, 4th Edition, CRC press, 2015.

REFERENCE BOOKS:

1. Shreiner and Angel, “Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL”, 6th Edition, Pearson, 2011.
2. Foley, Van Dam, Feiner, & Hughes, Addison-Wesley, “Computer Graphics: Principles and Practice”,1st Edition, Pearson Education India, 2013.
3. John Vince “Mathematics for Computer Graphics”, 5th Edition, Springer, 2017.
4. Hearn, Baker and Carithers, “Computer Graphics with OpenGL”,4th Edition, Pearson, 2015.
5. Evan M. Hahn,“Express in Action”, 1st Edition, Manning Publications, 2014.

SKILLS:

- ✓ *Utilization of GPU during graphics applications*
- ✓ *Usage of modern 3D graphics library.*
- ✓ *Produce usable graphics user-interfaces.*
- ✓ *Manipulate 3D objects in virtual environments.*

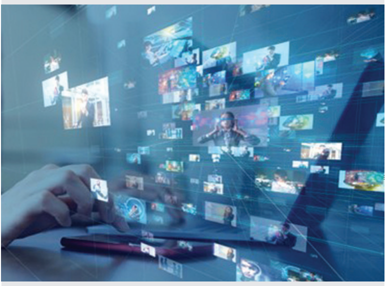


Image Source: https://d1m75rqqgidzqn.cloudfront.net/wp-data/2020/09/28191405/iStock 1194783078.jpg

22IT815 DIGITAL IMAGE PROCESSING

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Relationships between pixels, Color image fundamentals, Fourier Transform

COURSE DESCRIPTION AND OBJECTIVES:

It builds image processing approach to become familiar with digital image fundamentals. To get exposed to simple image enhancement techniques in Spatial and Frequency domain. To learn concepts of degradation function and restoration techniques. To study the image segmentation and representation techniques. To become familiar with image compression and recognition methods. This course know and understand the basics and fundamentals of digital image processing, such as digitization, sampling, quantization, and 2D-transforms. It operates on images using the techniques of smoothing, sharpening and enhancement. It uses the restoration concepts and filtering techniques and illustrate the basics of segmentation

MODULE-1

UNIT-1 **10L+6T+0P=16 Hours**

DIGITAL IMAGE FUNDAMENTALS:

Digital Image Fundamentals: Steps in Digital Image Processing, Components, Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Relationships between pixels, Color image fundamentals, RGB, HSI models, Two-dimensional mathematical preliminaries, 2D transforms - DFT, DCT.

UNIT-2 **6L+10T+0P=16 Hours**

IMAGE ENHANCEMENT, FREQUENCY DOMAIN AND IMAGE RESTORATION:

Image Enhancement: Spatial Domain: Gray level transformations, Histogram processing, Basics of Spatial Filtering– Smoothing and Sharpening Spatial Filtering.

Frequency Domain: Introduction to Fourier Transform, Smoothing and Sharpening frequency domain filters, Ideal, Butterworth and Gaussian filters, Homomorphic filtering, Color image enhancement.

Image Restoration: Image Restoration, degradation model, Properties, Noise models, Mean Filters, Order Statistics, Adaptive filters, Band reject Filters, Band pass Filters, Notch Filters, Optimum Notch Filtering, Inverse Filtering, Wiener filtering.

PRACTICES:

- Explain Steps in Digital Image Processing.
- Explain Image Sampling and Quantization.
- Discus on DFT, DCT.
- Explain Gray level transformations.
- Explain Fourier Transform for Smoothing and Sharpening.
- Discus on Image Restoration - degradation model.

MODULE-2

UNIT-1 **10L+6T+0P=16 Hours**

IMAGE SEGMENTATION:

Image Segmentation: Edge detection, Edge linking via Hough transform, Thresholding, Region based segmentation, Region growing, Region splitting and merging, Morphological processing, erosion and dilation, Segmentation by morphological watersheds, basic concepts, Dam construction, Watershed segmentation algorithm.

UNIT-2 **6L+10T+0P=16 Hours**

IMAGE COMPRESSION AND RECOGNITION:

Image Compression and Recognition: Need for data compression, Huffman, Run Length Encoding, Shift codes, Arithmetic coding, JPEG standard, MPEG. Boundary representation, Boundary description, Fourier Descriptor, Regional Descriptors, Topological feature, Texture Patterns and Pattern classes, Recognition based on matching.

PRACTICES:

- Explain Edge detection, Edge linking via Hough transform.
- Explain Morphological processing.
- Discus on Watershed segmentation algorithm.
- Discus on Need for data compression.
- Explain Arithmetic coding.
- Discus on JPEG standard, MPEG. Boundary representation.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply fundamental techniques of digital image acquisition, sampling, quantization, and color space models for image representation and analysis.	PO1, PO2, PO3, PO5, PSO1
CO2	Analyze image enhancement and restoration techniques in both spatial and frequency domains to improve image quality under various conditions.	PO1, PO2, PO4, PO5, PO6, PSO1, PSO2
CO3	Evaluate segmentation methods including thresholding, edge detection, morphological operations, and watershed algorithms for accurate image analysis.	PO1, PO2, PO4, PO5, PO10, PSO1, PSO3
CO4	Create compressed and interpretable image representations using coding standards and recognition techniques for intelligent pattern matching.	PO1, PO3, PO5, PO11, PO12, PSO1, PSO3

TEXT BOOK:

1. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 3rd Edition, Pearson, 2010.

REFERENCE BOOKS:

1. Kenneth R. Castleman, “Digital Image Processing”, 1st Edition, Pearson, 2006.
2. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins, “Digital Image Processing using MATLAB”, 2nd Edition, Pearson Education, 2011.
3. D,E. Dudgeon and RM. Mersereau, “Multidimensional Digital Signal Processing”, 1st Edition, Prentice Hall Professional Technical Reference, 1990.
4. William K. Pratt, “Digital Image Processing”, 1st Edition, John Wiley, New York, 2002.

SKILLS:

- ✓ *Develop Image Reconstruction using rotation and scaling approaches.*
- ✓ *Apply Probability and Random Variable Functions, Conditional and Joint Probability Density Functions for image analysis.*
- ✓ *Try to analyse of Thresholding -Region based segmentation.*
- ✓ *To use Fourier Transform for Smoothing and Sharpening frequency domain filters of image.*
- ✓ *Analyse various image coding such as Huffman coding, Run Length Encoding, Shift codes, Arithmetic coding,*

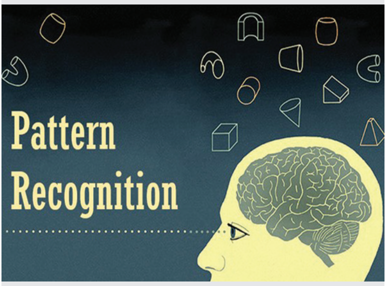


Image source: Paint3D (Created by own from website - https://dornsife.usc.edu/news/stories/2566/pattern-recognition/)

22IT828 PATTERN RECOGNITION

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Basics of data mining, Probability & statistics, Python & R programming.

COURSE DESCRIPTION AND OBJECTIVES:

Pattern recognition theory and practice is concerned with the design, analysis, and development of methods for the classification or description of patterns, objects, signals, and processes. The objective of this course is to enable the student to understand the data predictions in various fields.

MODULE-1

UNIT-1 **8L+6T+0P=14 Hours**

INTRODUCTION TO PATTERN RECOGNITION:

Fundamental Concepts of Pattern Recognition: Pattern Recognition, Types of Pattern Recognition algorithms, Scope of Pattern Recognition, Design principles and blocks of a typical pattern recognition system, Advantages and Importance of Pattern Recognition, Applications of Pattern Recognition, Types of Learning and Adaption. Mathematical representations – Probability theory, Decision theory, vector space and algebra.

UNIT-2 **8L+10T+0P=18 Hours**

BAYESIAN LEARNING AND PARAMETER ESTIMATION METHODS:

Bayes Decision Theory: Minimum-error-rate classification, Classifiers, Discriminant functions, Decision surfaces. Normal density and Discriminant functions, Discrete features.

Parameter Estimation Methods: Maximum-Likelihood estimation, Maximum Posteriori estimation, Bayesian estimation. Gaussian mixture models, Expectation, Maximum entropy estimation.

Applications of Sequential Pattern Recognition – Hidden Markov Models (HMMs), Discrete HMMs; Continuous HMMs. Nonparametric techniques for density estimation – Parzen-window method; K-Nearest Neighbour method.

PRACTICES:

- During the day, a clock at random stops once at any time. If x be the time when it stops and the PDF for x is given by:

$$f(x) = \begin{cases} 1/24, & \text{for } 0 \leq x \leq 240 \\ 0, & \text{otherwise} \end{cases}$$

  Calculate the probability that clock stops between 2 pm and 2:45 pm.
- Using R, produce a plot of the likelihood functions.
- Write a R script to implement Hidden Markov Models (HMMs) by using HMM package and Predicting the state by building the HMM model.
- Find the nearest neighbors between two sets of data using the unsupervised algorithms within sklearn.neighbors in python.
- Develop a python script to cluster data by using Scikit Gaussian Mixture class by building Gaussian Mixture model (GMM).

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

FEATURE MODELLING AND DIMENSIONALITY REDUCTION:

Feature identification, selection, and extraction: Distance measures, clustering transformation and feature ordering, clustering in feature selection, feature selection through maximization and approximations.

Dimensionality reduction: Fisher Discriminant analysis; Principal component analysis; Factor Analysis.

Non-metric methods for pattern classification: non-numeric data or nominal data. Decision trees: Classification and Regression Trees (CART).

UNIT-2 **8L+8T+0P=16 Hours**

PATTERN CLASSIFICATION METHODS AND APPLICATIONS:

Linear Discriminant functions: Gradient descent procedures, Perceptron, Support vector machines a brief introduction.

Artificial neural networks: Multilayer Perceptron – feed forward neural network. A brief introduction to deep neural networks, convolution neural networks, recurrent neural networks.

Applications of statistical and neural network – based pattern classifiers in speech recognition, image recognition and target recognition.

PRACTICES:

- Load the Pima Indians Diabetes dataset in R that contains several biological attributes from medical reports. Create a correlation matrix from these attributes and identify the highly correlated attributes. Remove the age attribute as it correlates highly (> 0.75) with the pregnant attribute.
- For the dataset in the above example, construct a Learning Vector Quantization (LVQ) model in R using varImp to estimate the variable importance.
- Write a R script to improve feature selection process with Principal Component Analysis (PCA) on any built-in dataset.
- Write a R script to study a heart disease data set and to model a support vector machine (SVM) classifier for predicting whether a patient is suffering from any heart disease or not.
- Write a R script to visualize the second cat from the training dataset directly using the Viewer pane.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the fundamental principles and mathematical foundations of pattern recognition including probability, decision theory, and vector algebra.	PO1, PO2, PO3, PO5, PO12, PSO1
CO2	Analyze Bayesian learning models and parameter estimation techniques including HMMs and non-parametric methods for effective pattern modeling.	PO1, PO2, PO4, PO5, PO6, PO12, PSO1, PSO2
CO3	Evaluate dimensionality reduction and feature selection strategies using PCA, LDA, and clustering for pattern discrimination.	PO1, PO2, PO4, PO5, PO12, PSO1, PSO3
CO4	Create classification models using SVMs, ANN, CNNs, and apply them to real-world pattern recognition tasks like speech, image, and target analysis.	PO1, PO3, PO5, PO11, PO12, PSO1, PSO3

SKILLS:

- ✓ *Recognizing and classifying concepts.*
- ✓ *Identifying solutions to problems.*
- ✓ *Developing autonomy in the thinking process.*
- ✓ *Working with Other People to find creative solutions.*

TEXT BOOKS:

1. J.I. Tou & R.C. Gonzalez, “Pattern Recognition Priciples”, 1st Edition, Addition-Wesley, 2014.
2. R. Schalkoff, “Pattern Recognition - Statistical, Structural and Neural Approaches”, John Wiley, 1992.

REFERENCE BOOKS:

1. P.A. Devijer & J. Kittler, “Pattern Recognition - A Statistical Approach”, 2nd Edition, Prentice-Hall, 2009.
2. Christopher. M. Bishop, “Pattern recognition and machine learning”, Springer, 2006.
3. M. NarasimhaMurthy and Der V Susheela Devi, “Introduction to Pattern Recognition and Machine Learning” IISC Press.
4. R.O.Duda, P.E.Hart and D.G.Stork, “Pattern Classification”, John Wiley, 2001
5. S.Theodoridis and K.Koutroumbas, “Pattern Recognition”, 4th Edition, Academic Press, 2009.
6. K Fukunaga, “Statistical pattern Recognition”, 1st Edition, Academic Press, 2000.

22IT819 IMAGE AND VIDEO ANALYTICS

Image source: https://www.einfochips.com/blog/role-video-analytics-smart-city-traffic-control/

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Basic ML, Image processing

COURSE DESCRIPTION AND OBJECTIVES:

This course explores and demonstrate real time image and video analytics in solving practical problems of commercial and scientific interests. The objective of this course is to offer the knowledge and skill on the basic principles and concepts in digital image and video processing.

MODULE-1

UNIT-1 **12L+0T+8P=20 Hours**

IMAGE PROCESSING:

Python Installation and Working of it, get familiar with python variables and data types, Operator understanding and its usage, detail study of python blocks, Hands on with conditional blocks using if, else and elif

UNIT-2 **12L+0T+8P=20 Hours**

FEATURE EXTRACTION AND TEXTURE ANALYSIS:

Feature Extraction: Binary object feature, Histogram-based (Statistical) Features, Intensity features, Shape feature extraction, PCA - SIFT – SURF. Texture Analysis - Concepts and classification, statistical, structural and spectral analysis.

Object Recognition: Patterns and pattern class, Bayes' Parametric classification, Feature Selection and Boosting, Template- Matching. Content Based Image Retrieval - Feature based image retrieval, Object Based Retrieval.

PRACTICES:

- Video surveillance

  Description:

  Crowd management can provide an estimated crowd count in real time and trigger alerts when a threshold is reached or surpassed. They can also analyze crowd flow to detect movement in unwanted or prohibited directions.

- Traffic analysis

  Description:

  Vehicle counting, or differentiating between cars, trucks, buses, taxis, and so on, generates high-value statistics used to obtain insights about traffic. Installing speed cameras allows for precise control of drivers. Automatic license plate recognition identifies cars that commit an infraction or, thanks to real-time searching, spots a vehicle that has been stolen or used in a crime.

MODULE-2

UNIT-1 **12L+0T+8P=20 Hours**

VIDEO TRACKING:

Digital Video, Sampling of video signal, Video Enhancement and Noise Reduction- Rate control and buffering, MPEG, H.264, Inter frame Filtering Techniques, Fundamentals of Motion Estimation and Motion Compensation.

SKILLS:

- ✓ *Able to apply various fundamentals for image processing.*
  .
- ✓ *Able to implement video analysis action recognition.*
- ✓ *Able to identify and recognize object in motion.*

Change Detection, Background modelling, Motion Segmentation, Simultaneous Motion Estimation and Segmentation, Motion Tracking, Multi-target/Multi-camera tracking.

UNIT-2 **12L+0T+8P=20 Hours**

VIDEO ANALYSIS ACTION RECOGNITION:

Video Analysis Action Recognition, Video based rendering, Context and scene understanding. Case Study: Surveillance - Advanced Driver Assistance System.

PRACTICES:

- Smart Encoder Appliance.
- Self Service Portal.
- Smart Search Example: “person wearing blue”
- Video Summarization.
- Suspect Identification.
- Real-Time Alerts.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply Python programming constructs to perform image processing tasks including variable usage, operators, conditional blocks, and basic operations.	PO1, PO2, PO3, PO5, PSO1
CO2	Analyze image features and textures using techniques like PCA, SIFT, SURF, and apply classification methods for object recognition and retrieval.	PO1, PO2, PO4, PO5, PO6, PSO1, PSO3
CO3	Evaluate motion estimation, background modeling, and filtering techniques for effective multi-object video tracking and segmentation.	PO1, PO2, PO4, PO5, PO10, PO11, PSO1, PSO2
CO4	Create context-aware video analytics systems for action recognition and scene understanding using case studies such as surveillance and ADAS.	PO1, PO3, PO5, PO11, PO12, PSO1, PSO3

TEXT BOOKS:

1. Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, 3rd Edition., Prentice-Hall, 2010.
2. Murat Tekalp, “Digital Video Processing”, 2nd Edition, Prentice Hall, 2015.

REFERENCE BOOKS:

1. Oge Marques,” Practical Image and Video Processing Using MATLAB”, Wiley-IEEE Press,2011
2. Yu Jin Zhang, “Image Engineering: Processing, Analysis and Understanding”, Tsinghua University Press, 2009.
3. Mark Nixon and Alberto S. Aquado, “Feature Extraction & Image Processing for Computer Vision”, 3rd Edition, Academic Press, 2012.
4. Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer, 2010.
5. Boguslaw Cyganek, “Object Detection and Recognition in Digital Images: Theory and Practice”, Wiley 2013.

22IT810 COMPUTER VISION

Image source: https://cdn.nextgov.com/media/img/cd/2020/11/03/NGai20201103/860x394.jpg

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Artificial intelligence and Machine learning.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the development of algorithms and techniques to analyze and interpret the visible world around us. This requires an understanding of the fundamental concepts related to multi-dimensional signal processing, feature extraction, pattern analysis visual geometric modeling, stochastic optimization, etc. Knowledge of these concepts is necessary for this field, to explore and contribute to research and further developments in the field of computer vision. Applications range from Biometrics, Medical diagnosis, document processing, and mining of visual content, to surveillance, advanced rendering, etc.

MODULE-1

UNIT-1 **12L+0T+6P=18 Hours**

COMPUTER VISION AND TRANSFORMATION:

Computer Vision: Introduction to computer vision, Digital Image Formation and low-level processing, Overview and State-of-the-art, Fundamentals of Image Formation.

Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution, Filtering, Image Enhancement, Restoration, Histogram Processing.

Image Formation: Geometric primitives and transformations, 2D transformations, 3D transformations, 3D rotations, 3D to 2D projections.

UNIT-2 **12L+0T+10P=22 Hours**

PHOTOMETRIC IMAGE FORMATION, DIGITAL CAMERA AND IMAGE PROCESSING:

Photometric Image Formation: Lighting, Reflectance and shading, Optics.

The Digital Camera: Sampling and aliasing, Color, Compression.

Image Processing: Point operations- Pixel transforms, Color transforms, Compositing and matting, Histogram equalization. Linear filtering – Separable filtering, Band-pass and steerable filters. More neighbourhood operators – nonlinear filtering, Bilateral filtering, Binary image processing. Fourier transforms, Pyramids and wavelets, and Geometric transformations. Model fitting and optimization.

PRACTICES:

- Digital image formation and low level processing.
- Orthogonal, Euclidean, Affine and Projective.
- Filtering, Image Enhancement, Restoration, Histogram Processing.
- 2D transformations, 3D transformations.
- 3D rotations, 3D to 2D projections.
- Reflectance and shading.
- Color and compression.
- Pixel transforms, Color transforms.
- Histogram equalization, Linear filtering.
- Fourier transforms, Pyramids and wavelets.

SKILLS:

- ✓ Develop image analysis algorithms.
- ✓ Develop Deep Learning frameworks to solve problems of computer vision.
- ✓ Design and create platforms for image processing and visualization.

MODULE-2

UNIT-1 **12L+0T+6P=18 Hours**

RECOGNITION, FEATURE DETECTION AND MATCHING, IMAGE ALIGNMENT:

Recognition: Instance recognition, Image classification, Object detection, Semantic segmentation, Video understanding, Vision and language.

Feature Detection and Matching: Points and Patches, Edges and contours, Contour tracking, Line and vanishing points, and segmentations.

Image Alignment and Stitching: Pairwise alignment, Image stitching, Global alignment, and Compositing.

Motion Estimation: Translational alignment, Parametric motion, Optical flow, Layered motion.

UNIT-2 **12L+0T+10P=22 Hours**

COMPUTATIONAL PHOTOGRAPHY:

Computational Photography: Photometric calibration, High dynamic range imaging, Super resolution denoising and blur removal, Image matting and compositing, Texture analysis and synthesis.

Depth Estimation: Epipolar geometry, Sparce correspondence, Dense correspondence, Local methods, Global optimization, Depth neural network.

3D Reconstruction: Shape form X, 3D scanning, Surface representation, Point based representation.

PRACTICES:

- Image classification, Object detection.
- Points and Patches, Edges and contours.
- Global alignment and compositing.
- Parametric motion and optical flow.
- Photometric calibration, High dynamic image range.
- Super resolution denoising.
- Epipolar geometry.
- Global optimization.
- 3D scanning, Surface representation.
- Point based representation.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply concepts of digital image formation, 2D/3D transformations, filtering, and enhancement techniques in computer vision applications.	PO1, PO2, PO3, PO5, PSO1
CO2	Analyze photometric image formation, color theory, filtering, and model fitting techniques in the context of digital cameras and real-world image data.	PO1, PO2, PO4, PO5, PO6, PSO1, PSO2
CO3	Evaluate feature detection and matching methods, motion estimation, object recognition, and image alignment for high-level vision tasks.	PO1, PO2, PO4, PO5, PO10, PO11, PSO1, PSO3
CO4	Create computational photography systems involving HDR imaging, super-resolution, depth estimation, and 3D reconstruction using optimization and neural models.	PO1, PO3, PO5, PO11, PO12, PSO1, PSO3

TEXT BOOK:

1. Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer-Verlag London Limited, 2021.

REFERENCE BOOKS:

1. Richard Hartley and Andrew Zisserman, “Multiple View Geometry in Computer Vision”, 2nd Edition, Cambridge University Press, March 2004.
2. Christopher M. Bishop; Pattern Recognition and Machine Learning, Springer, 2006.
3. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992.
4. K. Fukunaga; Introduction to Statistical Pattern Recognition, 2nd Edition, Academic Press, Morgan Kaufmann, 1990.

Image source: https://cdn-images-1.medium.com/max/670/1*Q3Lby1Un7SBDEHYzxtsOEw.png

22IT820 MEDICAL IMAGE ANALYSIS

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Image processing, joint probability.

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the medical image terminology, medical image modalities, image reconstruction methods, biomedical image analysis, segmentation, image representation and classification.

MODULE-1

UNIT-1 **12L+0T+4P=16 Hours**

BASICS OF CONTINUOUS TIME SIGNALS AND SYSTEMS:

Medical Imaging: A Collaborative Paradigm, Medical Imaging Modalities, Medical imaging from Physiology to Information Processing, Understanding Physiology and Imaging Medium, Data Acquisition and Image Reconstruction, Image Analysis and Applications.

Image Formation: Image Coordinate System, 2-D Image Rotation, 3-D Image Rotation and Translation Transformation, Linear Systems, Point Source and Impulse Functions, Probability and Random Variable Functions, Conditional and Joint Probability Density Functions, Independent and Orthogonal Random Variables, Image Formation, PSF and Spatial Resolution, Signal-to-Noise Ratio, Contrast-to-Noise Ratio, Pin-hole Imaging,

UNIT-2 **12L+0T+12P=24 Hours**

LTI SYSTEM AND FOURIER SERIES:

Medical Imaging Modalities: X-Ray Imaging: X-Ray Imaging, X-Ray Generation, X-Ray 2-D Projection Imaging, X-Ray Mammography, X-Ray CT, Spiral X-Ray CT, Contrast Agent, Spatial Resolution, and SNR,

Medical Imaging Modalities: Magnetic Resonance Imaging: MRI Principles, MR Instrumentation, MRI Pulse Sequences, Spin-Echo Imaging, Inversion Recovery Imaging, Echo Planar Imaging, Gradient Echo Imaging, Flow Imaging, fMRI, Diffusion Imaging, Contrast, Spatial Resolution, and SNR.

PRACTICES:

- Image sampling and quantization.
- Analysis of spatial and intensity resolution of images.
- Intensity transformation of images.
- DFT analysis of images.
- Transforms (Walsh, Hadamard, DCT, Haar)
- Histogram Processing.

MODULE-2

UNIT-1 **12L+0T+4P=16 Hours**

ANALYSIS IN FREQUENCY DOMAIN:

Medical Imaging Modalities: Ultrasound Imaging: Propagation of Sound in a Medium, Reflection and Refraction, Transmission of Ultrasound Waves in a Multilayered Medium, Attenuation, Ultrasound Reflection Imaging, Ultrasound Imaging Instrumentation, Imaging with Ultrasound: A-Mode, M-Mode, B-Mode, Doppler Ultrasound Imaging, Contrast, Spatial Resolution, and SNR.

Image Reconstruction: Radon Transform and Image Reconstruction, The Central Slice Theorem, Inverse Radon Transform, Back projection Method, Iterative Algebraic Reconstruction Methods, Estimation Methods, Fourier Reconstruction Methods, Image Reconstruction in Medical Imaging Modalities, Image Reconstruction in X-Ray CT, Image Reconstruction in Nuclear Emission Computed Tomography: SPECT and PET.

UNIT-2 **12L+0T+12P=24 Hours**

SAMPLING:

Adaptive Arithmetic Mean Filter, Image Sharpening and Edge Enhancement, Feature Enhancement Using Adaptive Neighborhood Processing, Frequency Domain Filtering, Wiener Filtering, Constrained Least Square Filtering, Low-Pass Filtering, High-Pass Filtering, Homomorphic Filtering, Wavelet Transform for Image Processing , Image Smoothing and Enhancement Using Wavelet Transform.

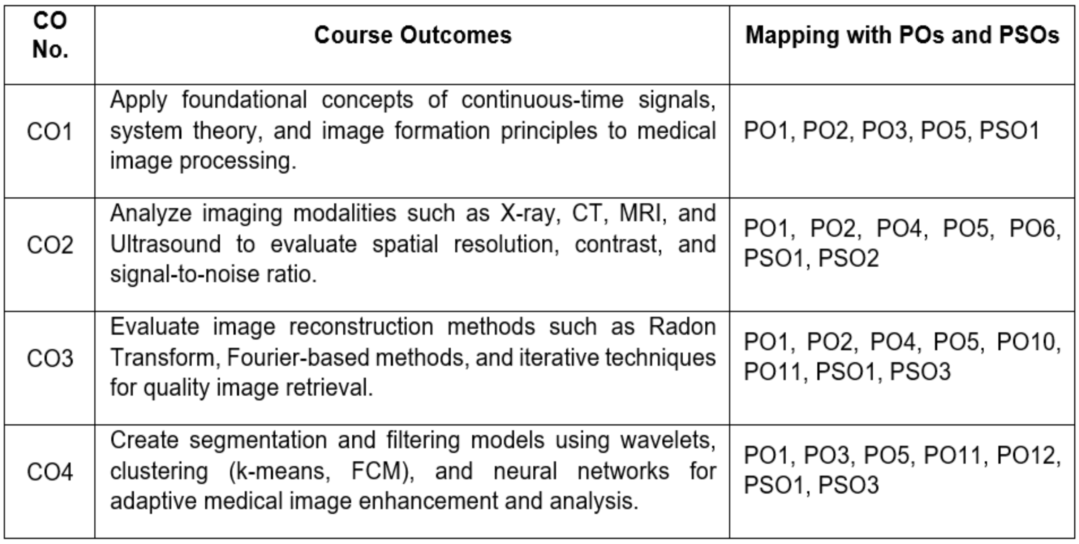
Image Segmentation: Pixel-Based Direct Classification Methods, Optimal Global Thresholding, Pixel Classification Through Clustering, Data Clustering, k-Means Clustering, Fuzzy c-Means Clustering, An Adaptive FCM Algorithm, Region-Based Segmentation, Region-Growing, Region-Splitting, Advanced Segmentation Methods , Estimation-Model Based Adaptive Segmentation, Image Segmentation Using Neural Networks, Backpropagation Neural Network for Classification, The RBF Network, Segmentation of Arterial Structure in Digital Subtraction Angiograms ,

PRACTICES:

- Image Enhancement-Spatial filtering.
- Image Enhancement- Filtering in frequency domain.
- Image segmentation – Edge detection, line detection and point detection.
- Basic Morphological operations.
- Basic Thresholding functions.
- Analysis of images with different color models.
- Analysis of medically acquired images.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply foundational concepts of continuous-time signals, system theory, and image formation principles to medical image processing.	PO1, PO2, PO3, PO5, PSO1
CO2	Analyze imaging modalities such as X-ray, CT, MRI, and Ultrasound to evaluate spatial resolution, contrast, and signal-to-noise ratio.	PO1, PO2, PO4, PO5, PO6, PSO1, PSO2
CO3	Evaluate image reconstruction methods such as Radon Transform, Fourier-based methods, and iterative techniques for quality image retrieval.	PO1, PO2, PO4, PO5, PO10, PO11, PSO1, PSO3
CO4	Create segmentation and filtering models using wavelets, clustering (k-means, FCM), and neural networks for adaptive medical image enhancement and analysis.	PO1, PO3, PO5, PO11, PO12, PSO1, PSO3

TEXT BOOK:

1. Atam P. Dhawan, “Medical Image Analysis”, 2nd Edition, John Wiley, 2011

REFERENCE BOOK:

1. Klaus D. Toennies, “Guide to Medical Image Analysis Methods and Algorithms”, Springer London, 2012.

SKILLS:

- ✓ *Develop Image Reconstruction using rotation and scaling approaches.*
- ✓ *Apply Probability and Random Variable Functions, Conditional and Joint Probability Density Functions for image analysis.*
- ✓ *Try to analyse of Spin-Echo Imaging, Inversion Recovery Imaging, Echo Planar Imaging, Gradient Echo Imaging.*
- ✓ *Use various mode such as A-Mode, M-Mode, B-Mode for Ultrasound image.*
- ✓ *Analyse various image filtering such as Constrained Least Square Filtering, Low-Pass Filtering, High-Pass Filtering, Homomorphic Filtering.*

22IT803 ARTIFICIAL INTELLIGENCE

Source: https://cionews.co.in/artificial-intelligence-spending-to-reach-434-bn/

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Discrete mathematics, Probability and statistics.

COURSE DESCRIPTION AND OBJECTIVES:

The primary objective of this course is to introduce the basic principles, techniques, and applications of Artificial Intelligence. In addition to this, student will understand the building blocks of AI such as search, knowledge representation, inference, logic and learning. This course enables the students to develop a small AI system for real time problems.

MODULE-1

UNIT-1 **6L+6T +0P=12 Hours**

INTELLIGENT SYSTEMS:

Introduction, what is AI, Examples of AI systems, Brief history of AI, Intelligent Agent, Agents and environments, The concept of rationality, The nature of environments, Structure of agents.

UNIT-2 **10L+10T+0P=20 Hours**

PROBLEM SOLVING KNOWLEDGE REPRESENTATION:

State Space Problem; Searching: Uniform search, Informed Search: Solving problems by searching: Best First Search, A* algorithm, AO* algorithm, Heuristic functions, Hill climbing, searching game trees: Min Max Search, Alpha Beta pruning. Propositional logic: Logical agents, reasoning patterns in propositional logic, Inference in propositional logic, Resolution, forward chaining, Backward chaining.

PRACTICES:

- Implementation of Vacuum Cleaner Agent.
- Implementation of BFS, DFS.
- Implementation of 8-Puzzle problem using A* algorithm.
- Implementation of n queen’s problem.
- Implementation of MIN-MAX algorithm.
- Representation of simple fact for a statement.

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

PLANNING:

First order logic: Reasoning patterns in First order logic, Inference in First order logic, Resolution, forward chaining, Backward chaining, Unification. problem; The planning problem: Planning with state space search, Partial order planning, planning graphs, Planning with propositional logic, Analysis with planning approaches.

UNIT-2 **8L+8T+0P=16 Hours**

LEARNING:

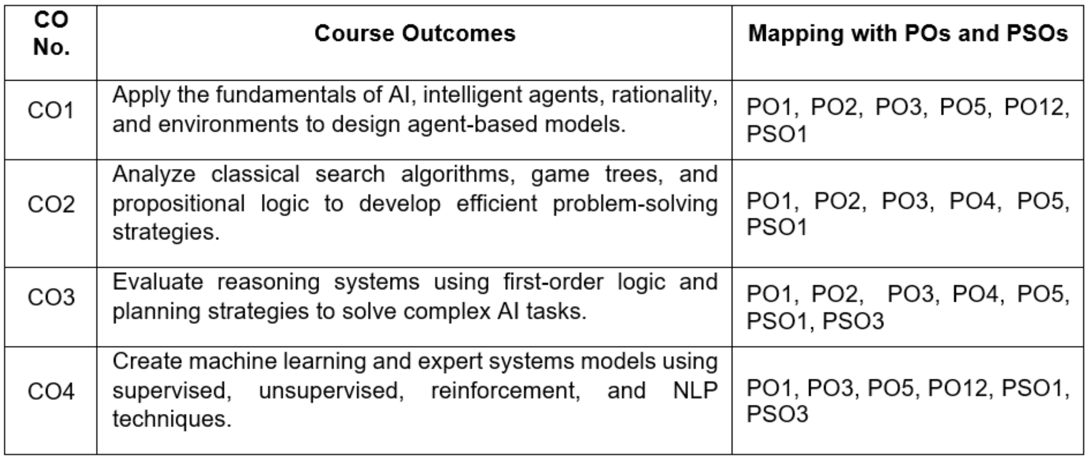
Forms of learning: Supervised Learning Unsupervised learning, Reinforcement learning, Ensemble Learning, Machine Learning, Learning Decision Trees, Expert system, Natural Language Processing.

PRACTICES:

- Implementation of a graph and apply BFS on it.
- Implementation on backward and forward reasoning.
- Implementation of containing facts related to following predicates.
- Location (city, state) Stays (person, city) Display:
- list of persons, state and city.
- (ii) Given person, find the state in which he is staying.
- Implementation of the answers about family members and relationships. Include predicates & clauses which define sister, brother, father, mother, Grandchild, grandfather and uncle. The program should be able to answer question such as following.
  - o Father (X, bob)
  - o Grandson (X, Y)
  - o Uncle (bill, Sue)
  - o Mother (marry, X)
- Implementation of an inductive learning algorithm for decision trees.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the fundamentals of AI, intelligent agents, rationality, and environments to design agent-based models.	PO1, PO2, PO3, PO5, PO12, PSO1
CO2	Analyze classical search algorithms, game trees, and propositional logic to develop efficient problem-solving strategies.	PO1, PO2, PO3, PO4, PO5, PSO1
CO3	Evaluate reasoning systems using first-order logic and planning strategies to solve complex AI tasks.	PO1, PO2, PO3, PO4, PO5, PSO1, PSO3
CO4	Create machine learning and expert systems models using supervised, unsupervised, reinforcement, and NLP techniques.	PO1, PO3, PO5, PO12, PSO1, PSO3

TEXT BOOKS:

1. Stewart Russell and Peter Norvig, “Artificial Intelligence-A Modern Approach”, 4th Edition, Pearson Education/ Prentice Hall, 2020.
2. Ivan Bratko, “Prolog Programming for Artificial Intelligence”, 4th Edition, Addison Welsey, 2011.

REFERENCE BOOKS:

1. George F Luger, “Artificial Intelligence”, 6th Edition, Pearson Education,2021.
2. Elaine Rich, Kevin Knight and Shiva Shankar B Nair, “Artificial Intelligence”, 3rd Edition, Tata McGraw, 2017
3. Dan W Patterson, “Introduction to Artificial Intelligence and Expert systems”, 2nd Edition Pearson, 2015.

SKILLS:

- ✓ *Analyze Intelligent systems.*
- ✓ *Apply problem solving techniques.*
- ✓ *Interface various knowledge representation.*
- ✓ *Create a dynamic planning.*

Image source https://www.educba.com/what-is-big-data-analytics/

22IT805 BIG DATA ANALYTICS

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Basics of database, Data mining

COURSE DESCRIPTION AND OBJECTIVES:

This course gives an overview of Big Data, i.e., storage, retrieval, and processing of big data. In addition, it also focuses on the “technologies”, i.e., the tools/algorithms that are available for storage, and processing of Big Data. It also helps a student to perform a variety of “analytics” on different data sets and to arrive at positive conclusions.

MODULE-1

UNIT-1 **12L+0T+8P=20 Hours**

INTRODUCTION TO BIG DATA ANALYTICS:

Introduction: Data, Characteristics of data and types of digital data, Sources of data, working with unstructured data, Evolution, and definition of big data, Characteristics and need of big data, and Challenges of big data

Big Data Analytics: Overview of business intelligence, Data Science, and analytics, Meaning and characteristics of big data analytics, Need for big data analytics, Classification of analytics, Challenges to big data analytics, Importance of big data analytics, Basic terminologies in the big data environment.

UNIT-2 **12L+0T+8P=20 Hours**

HADOOP, HDFS:

Introducing Hadoop, Need of Hadoop, Limitations of RDBMS, RDBMS versus Hadoop, Distributed computing challenges, History of Hadoop, Hadoop overview, the use cases of Hadoop, Hadoop distributors, HDFS, Processing data with Hadoop, managing resources and applications with Hadoop YARN, Hadoop ecosystem.

PRACTICES:

- Examples of digital data.
- Types of digital data.
- Real time examples of unstructured data.
- Real time examples of structured data.
- Real time examples of semi structured data.
- Real time challenges of Big data environment.
- Role of Data analyst in decision making process.
- Hardware support to process huge amount of data.
- Distributed environment vs. parallel environment.
- CAP theorem.
- BASE theorem.
- Evolution of Hadoop.
- Installation of Hadoop.
- Basic commands in Hadoop.
- YARN installation.
- Managing resources with YARN.

MODULE-2

UNIT-1 **12L+0T+8P=20 Hours**

MAPREDUCE, PIG, HIVE, SPARK:

Introduction to mapper, reducer, combiner, partitioner, searching, sorting, compression, real-time applications using MapReduce, combiner, partitioner, Introduction to Hive, Hive architecture, Hive data types, Hive file format, Pig versus Hive. The anatomy of pig, Pig on Hadoop, Pig philosophy, Use case for pig, ETL processing, Pig Latin overview, Data types in pig, Relational operators, Piggy bank, Introduction, features of spark, components of spark,

UNIT-2 **12L+0T+8P=20 Hours**

PROGRAMMING USING MAPREDUCE, PIG, HIVE, SPARK:

Programming using MapReduce, Matrix multiplication using MapReduce, and page rank algorithm using MapReduce.

Hive query language (HQL),Partitions and bucketing, RCFile Implementation, working with XML files, User-defined Function(UDF) in Hive, programming with Resilient Distributed Datasets (RDD), Running pig, Execution modes of pig, HDFS commands, Word count example using pig.

PRACTICES:

- Advantages and drawbacks of MapReduce.
- Modules of MapReduce.
- Installation MapReduce.
- Configuration of MapReduce.
- Installation Hive.
- Running and execution modes of Hive.
- Real time application of Hive.
- Installation PIG.
- Running and execution modes of PIG.
- Real time application of PIG.
- Installation Spark.
- Running and execution modes of Spark.
- Real time application of Spark.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply big data technologies including HDFS, YARN, MapReduce, Pig, and Hive to store, manage, and process large-scale structured and unstructured datasets.	PO1, PO2,PO3, PO4, PO5, PO12, PSO1, PSO3
CO2	Analyze the architectures and dataflow models of distributed frameworks for handling large-scale data.	PO1, PO2,PO3, PO4, PO5, PO12, PSO1, PSO3
CO3	Evaluate the performance, scalability, and effectiveness of big data tools and algorithms for real-time analytics and business intelligence.	PO1, PO2,PO3, PO4, PO5, PO12, PSO1, PSO3
CO4	Create custom big data workflows and programs using MapReduce, Pig, Hive UDFs, and Spark RDDs to address data processing and analytics problems.	PO1, PO2,PO3, PO4, PO5, PO12, PSO1, PSO3

SKILLS:

- ✓ *To build and maintain reliable, scalable, distributed systems with Apache Hadoop.*
- ✓ *To write Map-Reduce based Applications*
- ✓ *To design and build applications using Hive and Pig based Big data Applications*
- ✓ *To learn tips and tricks for Big Data use cases and solutions.*

TEXT BOOKS:

1. Seema Acharya and Subhashini Chellappan, “Big Data and Analytics”, 1st Edition, Wiley Publishers, 2015.
2. Holden Karau, Andy Konwinski, Patrick Wendell and Matei Zaharia, “Learning Spark”, 1st Edition, Oreilly, 2015.

REFERENCE BOOKS:

1. Anand Rajaraman and Jeffrey David Ullman, “Mining of Massive Datasets”, Cambridge, University Press, 2012.
2. Boris lublinsky, Kevin t. Smith and AlexeyYakubovich, “Professional Hadoop Solutions”, 1st Edition, Wiley, 2015.
3. Chris Eaton and Dirkderoosetal, “Understanding Big data”, 1st Edition, McGraw Hill, 2012.
4. Tom White, “HADOOP: The definitive Guide”, 1st Edition, O Reilly 2012.
5. Vignesh Prajapati, “Big Data Analytics with R and Hadoop”,1st Edition, Packet Publishing, 2013.

22IT813 DATA SCIENCE USING PYTHON

Image Source:
https://i.ibb.co/
Lg4nrd8/image2.png

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Programming in python

COURSE DESCRIPTION AND OBJECTIVES:

Data Science using Python course will guide to learn how to use the power of Python for Data Pre-Processing and analysis, Data exploration and Visualization.

MODULE-1

UNIT-1 **12L+0T+8P=20 Hours**

DATA SCIENCE & NUMPY:

Introduction to Data Science: Introduction to Data Science, Data Science Terminology, Data Science Process, Data Science Project Roles, Industrial applications of Data Science. Essential Python Libraries, NumPy Basics: Arrays and Vectorized Computation: The NumPy ndarray: A Multidimensional Array Object, Universal Functions: Fast Element-wise Array Functions, Data Processing Using Arrays, File Input and Output with Arrays, Linear Algebra, Random Number Generation.

UNIT-2 **12L+0T+8P=20 Hours**

PANDAS:

Getting Started with pandas: Introduction to pandas Data Structures, Essential Functionality, Summarizing and Computing Descriptive Statistics, Handling Missing Data, Hierarchical Indexing, Data Loading, Storage, and File Formats, Reading and Writing Data in Text Format, Binary Data formats, Interacting with HTML and Web APIs, Interacting with data bases.

PRACTICES:

- How arrays of data are handled in the Python language itself, and how NumPy improves on this.
- Attributes of arrays: Determining the size, shape, memory consumption, and data types of arrays, Indexing of arrays, Slicing of arrays, Reshaping of arrays, Joining and splitting of arrays.
- NumPy's universal functions, compute summary statistics for the data, Broadcasting, Exploring Fancy Indexing, algorithms related to sorting values in NumPy arrays, use of NumPy's structured arrays and record arrays.
- Data Manipulation with Pandas: Introducing Pandas Objects, Data Indexing and Selection, Operating on Data in Pandas, Handling Missing Data, Hierarchical Indexing, Combining Datasets: Concat and Append, Merge and Join.

MODULE-2

UNIT-1 **12L+0T+8P=20 Hours**

PLAYING AND PLOTTING DATA:

Combining and Merging Data sets, Reshaping and Pivoting, Data Transformation, String Manipulation.

Plotting & Visualization: Matplotlib introduction, Plotting functions in pandas, Plotting Maps: Case Study for different Applications.

SKILLS:

- ✓ *Apply data science methods and techniques, and acquire analysis skills.*
- ✓ *Data manipulation and data cleaning using python and pandas regular expressions.*

UNIT-2 **12L+0T+8P=20 Hours**

DATA AGGREGATION AND GROUP OPERATIONS & TIME SERIES:

Data Aggregation and Group Operations: Group by Mechanics, Data Aggregation, Group-wise Operations and Transformations, Pivot Tables and Cross-Tabulation.

Time Series: Date & Time Data Types and Tools, Time series Basics, Date Ranges, Frequencies, and Shifting. Time zone Handling, Periods and Period Arithmetic, Resampling and Frequency Conversion, Time series plotting, Moving window functions, Performance, and memory usage notes.

PRACTICES:

- Data Aggregations on Multi-Indices: Panel Data, Concat and Append, Merge and Join, Aggregation and Grouping, Pivot Tables, Vectorized String Operations, Working with Time Series, High-Performance Pandas: eval() and query().
- Visualization with Matplotlib: Simple Line Plots, Scatter Plots, Visualizing Errors, Density and Contour Plots, Histograms, Binnings, and Density, Customizing Plot Legends, Customizing Colorbars, Multiple Subplots, Text and Annotation, Customizing Ticks, Customizing Matplotlib: Configurations and Stylesheets, Three-Dimensional Plotting in Matplotlib.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply Python libraries such as NumPy and pandas to load, manipulate, and process structured data for analysis.	PO1, PO2, PO3, PO5, PO12, PSO1
CO2	Analyze and visualize data using data wrangling techniques, transformation operations, and basic plots to derive insights.	PO1, PO2, PO3, PO5, PO12, PSO1, PSO3
CO3	Evaluate data quality using statistical measures, handle missing data, and use aggregation and grouping to interpret trends in large datasets.	PO1, PO2, PO3, PO5, PO6, PO12, PSO1, PSO3
CO4	Create time-series models with shifting, resampling, and rolling functions to extract temporal patterns and predict trends.	PO1, PO3, PO5, PO12, PSO1, PSO3

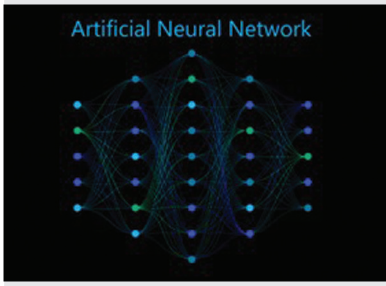
TEXT BOOK:

1. Wes McKinney, “Python for Data Analysis” 3rd Edition, O Reilly, 2022.

REFERENCE BOOKS:

1. Jake VanderPlas, “Python Data Science Handbook”, 1st Edition, O’Reilly, 2016.
2. Joel Grus, “Data Science from Scratch”, 2nd Edition, O Reilly, 2019.
3. Foster Provost & Tom Fawcett, “Data Science for Business”, O Reilly, 2013.

22IT804 ARTIFICIAL NEURAL NETWORKS



Source: https://media.istockphoto.com/vectors/neural-network-model-with-thin-synapses-between-neurons-vector-id1156059931?k=20&m=1156059931&s=612x612&w=0&h=UEkHaO0BFXlsHv6PDHxzKgo2fKJexkqtBDtZRAyw3ml=

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Basics of algorithms and programming

COURSE DESCRIPTION AND OBJECTIVES:

The course introduces the principles of neuro-computing with artificial neural networks, which widely used for addressing real-world problems such as classification, regression, pattern recognition, data mining, time-series modelling, etc.. This course majorly covers two kinds of learning such as supervised and unsupervised.

MODULE-1

UNIT-1 **8L+8T+0P=16 Hours**

ARCHITECTURE OF NEURAL NETWORK:

Introduction to Artifical Neural Networks: Introduction; Artificial neural networks; Historical development of neural networks; Biological Neural Networks; Basic building blocks; Terminologies. Fundamental Models of Artificial Neural Networks: Introduction, McCullochPitts Neuron model; Perceptron Networks: Introduction; Single layer perceptron; Brief introduction to multilayer perceptron networks.

UNIT-2 **8L+8T+0P=16 Hours**

MODELS OF ARTIFICIAL NEURAL NETWORKS:

Comparison between them and the Computer; Comparison between artificial and biological neural network; Fundamental Models of Artificial Neural Networks: Learning rules - Hebbian learning rule, perceptron learning rule, Delta learning rule, Widrow-H off rule or least mean square (LMS) rule, Competitive learning rule; Out star learning; Boltzmann based learning; Hebb net.

PRACTICES:

- Plot a few activation functions that are being used in neural networks.
- Generate ANDNOT function using McCulloch-Pitts neural net.
- Generate XOR function using McCulloch-Pitts neuron.
- With a suitable example demonstrate the perceptron learning law with its decision regions.

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

TYPES OF NEURAL NETWORKS:

Adaline and Madaline Networks: Introduction, Adaline, Madaline. Associative Memory Networks: Introduction; Feedback Networks: Introduction; Feed Forward Networks: Introduction; Self Organizing Feature Map: Introduction; Adaptive Resonance Theory: Introduction, ART Fundamentals, ART1, ART2.

UNIT-2 **8L+8T+0P=16 Hours**

ADVANCED NEURAL NETWORKS:

Adaline and Madaline Networks: Algorithms for pattern association, Hetero associative memory neural networks, Auto associative memory network, Bi-directional associative memory. Feedback Networks: Discrete Hopfiled Net, Continuous Hopfiled Net, Relation between BAM and Hopfiled Nets. FEED Forward Networks: Back Propagation Network (BPN), Radial Basis Function Network (RBFN). SelfOrganizing

Feature Map: Introduction; Methods used for determining the winner; Kohonen self-organizing feature maps; Learning vector quantization(LVQ); MaxNet, Maxican Hat, Hamming Net.

SKILLS:

- ✓ *Create different neural networks of various architectures both feed forward and feed backward.*
- ✓ *Perform the training of neural networks using various learning rules.*
- ✓ *Testing of neural networks and do the perform analysis of these networks for various pattern recognition applications.*

PRACTICES:

- Program to show Back Propagation Network for XOR function with Binary Input and Output.
- Program to show Back Propagation Network for XOR function with Bipolar Input and Output.
- Program to illustrate ART neural network.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply different neural networks such as MLP, SOM, Hopfield net, and ART, along with linear and nonlinear models for learning the data.	PO1, PO2, PO3, PO5, PO9, PO10, PSO1, PSO3
CO2	Analyze the differences between networks for supervised and unsupervised learning.	PO1, PO2, PO3, PO4, PO5, PO9, PO12, PSO1, PSO3
CO3	Evaluate the performance of various neural networks on different kinds of data.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1, PSO3
CO4	Design neural network models to classify/cluster data for achieving higher performance.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3

TEXT BOOK:

1. Sivanandam, S.Sumathi and S.N.Deepa; “Introduction to Neural Networks”, 2nd Edition., TATA McGraw HILL: 2005.

REFERENCE BOOKS:

1. Simon. Hhaykin, “Neural networks A comprehensive foundation”, 2nd Edition, Pearson Education, 2004.
2. B. Yegnanarayana, “Artificial neural networks”, 1st Edition., Prentice Hall of India Pvt. Ltd, 2005.
3. Li Min Fu, “Neural networks in Computer intelligence”,1st Edition., TMH, 2003.

22IT814 DEEP LEARNING

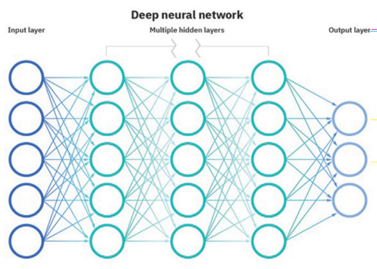


Image source: https://1.cms.s81c.com/sites/default/files/2021-01-06/ICLH_Diagram_Batch_01_03-DeepNeuralNetwork-WHITEBG.png

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Basics of Data Mining, Artificial Neural Networks, Python & R Programming

COURSE DESCRIPTION AND OBJECTIVES:

The objective of this course is to introduce major deep learning algorithms, the problem settings, and their applications to solve real world problems. The course also requires students to implement programming assignments related to these topics such as recurrent neural networks, long short-term memory cells and convolution neural networks.

MODULE-1

UNIT-1 **12L+0T+8P=20 Hours**

FUNDAMENTAL CONCEPTS OF DEEP LEARNING:

Introduction to Deep Learning: Deep Learning, Paradigms of learning problems, Types of Deep Neural Networks – Artificial Neural Network, Multi-layer neural network, Convolution neural network, Recurrent neural network, Deep belief network, Perspectives, and Issues in deep learning framework.

Applications of Deep Learning – Fraud detection, Customer relationship management systems, Computer vision, Natural language processing, Autonomous vehicles, E-commerce.

UNIT-2 **12L+0T+8P=20 Hours**

TYPES OF DEEP NEURAL NETWORKS:

Convolution Neural Networks – Basic structure of Convolutional Network, Case studies: Alex net, VGGNet, GoogLeNet, Applications of CNN– Object Detection, Content based image Retrieval.

Recurrent Neural Networks – Basic structure of RNN, Case studies: Long Short-Term Memory, Gated Recurrent Units, Back propagation through time, Bidirectional LSTMs, and Bidirectional RNNs, Applications of RNN– Language Modelling and Generating Text, Text Summarization.

PRACTICES:

- Develop a deep learning model with an Artificial Neural Network to create a predictive model based on the Churn dataset for new customers. The predictive model must predict for any new customer that he or she will stay in the bank or leave the bank. So that bank can offer something special for the customers whom the predictive model predicts will leave the bank.
- Develop a deep learning model of your First Image Processing Project with Convolutional Neural Network.
- Develop a deep learning model for fruit classification using GoogleNet Convolutional Neural Network.
- Develop a deep learning model to solve sequence prediction problem using RNN with Dropout condition.
- Develop a deep learning model to perform anomaly detection model using LSTM.

SKILLS:

- ✓ *To understand the fundamentals of deep learning.*
- ✓ *Main techniques in deep learning and the main research in this field.*
- ✓ *Design and implement deep neural network systems.*
- ✓ *Identify new application requirements in the field of computer vision.*

MODULE-2

UNIT-1 **12L+0T+8P=20 Hours**

GENERATIVE & ATTENTION MODELS:

Generative models – Restrictive Boltzmann Machines (RBMs), Deep Boltzmann Machines, Auto encoders, and Generative Adversarial Networks.

Attention and memory models – Attention Mechanism, Types of Attention models – Local Attention, Global Attention, and Self-Attention.

UNIT-2 **12L+0T+8P=20 Hours**

BETTER TRAINING OF DEEP NEURAL NETWORKS:

Training Neural Network – Keras, Tensorflow, Regularizations, Training deep models, Dropouts, Multi-task learning, Early Stopping, Parameter Initialization and Parameter Sharing, Newer optimization methods for neural networks (Adagrad, adadelta, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization).

PRACTICES:

- Develop a project for implementing Image Segmentation using CNN with Tensorflow.
- Develop a project for implementing Time Series Forecasting with LSTM Neural Network Python.
- Develop a project for building a Review Classification Model using Gated Recurrent Unit.
- Develop a project for building CNN Image Classification Models for Real Time Prediction.
- Design a Deep Learning Project on Store Item Demand Forecasting.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply forward and backward propagation concepts in different types of Artificial Networks such as MLP, CNN, RNN and Apply Pre-Trained models to perform classification tasks.	PO1, PO2, PO3, PO5, PO9, PO10, PO12, PSO1, PSO3
CO2	Evaluate the differences among MLP, CNN, and RNN functioning and prioritize the models along with different parameters (learning rate, activation function, batch size, epochs, optimizers, etc.) for different problems.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12, PSO1, PSO3
CO3	Analyze the relationship and limitations of classification, different model evaluation metrics.	PO1, PO2, PO3, PO5, PO9, PO10, PO12, PSO1, PSO3
CO4	Develop image and text classifier models for various real-time applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3

TEXT BOOKS:

1. Charu C Aggarwal, "Neural Networks and Deep learning", Springer International Publishing, 2018.
2. Satish Kumar, "Neural Networks, A Classroom Approach", Tata McGraw -Hill, 2007.

REFERENCE BOOKS:

1. P.A. Devijer & J. Kittler, "Pattern Recognition - A Statistical Approach", Prentice-Hall, 2008.
2. Christopher. M. Bishop, "Pattern recognition and machine learning", Springer, 2006.
3. M. NarasimhaMurthy and Der V Susheela Devi, "Introduction to Pattern Recognition and Machine Learning", IISC Press, 2008.

22IT824 NATURAL LANGUAGE PROCESSING

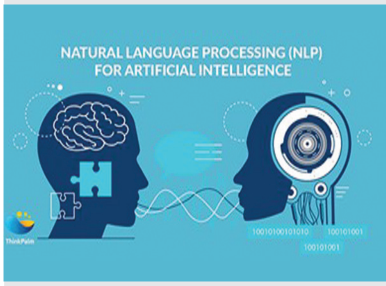

NATURAL LANGUAGE PROCESSING (NLP)
FOR ARTIFICIAL INTELLIGENCE

Image source: https://thinkpalm.com/wp-content/uploads/2019/04/BLOG_NLP-FOR-ARTIFICIAL-INTELLIGENCE_72-1.jpg

Hours Per Week :

L	T	P	C
3	-	2	3

PREREQUISITE KNOWLEDGE: Basics of classification techniques, statistical measures.

COURSE DESCRIPTION AND OBJECTIVES:

The goal of this course is to provide knowledge on fundamental concepts of Natural Language Processing. The methods for representing and classifying the written text is the primary focus. Students will learn tokenization, word representation (TF-IDF and word vectors) and classification techniques. Further understanding sentiment analysis and recommendation systems.

MODULE-1

UNIT-1 **8L+8T+0P=16 Hours**

N GRAM LANGUAGE MODELS:

Words, Corpora, Text Normalization, Minimum Edit distance. N Grams, Evaluating Language Models, Generalization and Zeros, Smoothing. Naïve Bayes and Sentiment Classification,

UNIT-2 **8L+8T+0P=16 Hours**

SENTIMENT ANALYSIS:

Training the Naïve Bayes classifier, Optimizing for sentiment analysis, Naïve Bayes for other text classification tasks, as a language model, Evaluation Test sets and Cross validation, Statistical significance of testing, advanced feature selection.

PRACTICES:

- Computing 1gram,2gram and N gram tokens.
- Stop word removing.
- Stemming and Lemmatization.
- Performing sentiment analysis using Naïve Bayes.
- Building a language model.
- Latent Semantic Analysis for feature selection/dimensionality reduction.
- Latent Dirichlet Allocation method for topic modelling.

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

VECTOR SEMANTICS AND SEQUENCE PROCESSING:

Vector Semantics: Lexical semantics, vector semantics, word vectors, Cosine for measuring similarity.

TF-IDF: weighting terms in the vector, Applications, Word2Vec, Visualizing Embeddings, semantic properties of embeddings, Evaluating Vector models.

Sequence Processing with Recurrent Networks: Simple Recurrent Networks, Applications of RNN's.

Deep Networks: Stacked and Bidirectional RNNs, managing Context in RNNs: LSTMs and GRU.

SKILLS:

- ✓ *Text pre-processing and Analyzing.*
- ✓ *Develop sentiment analyzers.*
- ✓ *Develop recommendation systems.*
- ✓ *Computing similarity score.*
- ✓ *Develop Context preserving language models.*

UNIT-2 **8L+8T+0P=16 Hours**

POS TAGGING:

Part of speech tagging: English Word classes, The Penn Treebank POS tag set, POS tagging, HMM POS tagging, Maximum Entropy Markov Models.

PRACTICES:

- Computing word similarity using cosine distance.
- Computing TF-IDF values for words in a corpus.
- Computing word embeddings using word2vec.
- Visualizing the word vectors using TSNE plots.
- Tokenizing and sequence generation.
- RNN based sentiment analysis.
- LSTM based sentiment analysis.
- GRU based sentiment analysis.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply foundational NLP techniques including tokenization, normalization, stemming, lemmatization, and construct language models using N-grams, Naïve Bayes classifiers, TF-IDF, and topic modeling.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO12, PSO1, PSO3
CO2	Analyze text data using sentiment analysis, part-of-speech tagging, and word embedding models like Word2Vec, and evaluate them with metrics such as accuracy, confusion matrix, ROC curves, and statistical validation techniques..	PO1, PO2, PO3, PO4, PO5, PO9, PO12, PSO1, PSO3
CO3	Evaluate the effectiveness of NLP models N-gram, Naïve Bayes, vector space, topic modeling, and deep learning (RNN, LSTM, GRU) by assessing their performance in classification, feature extraction, and sequence modeling tasks.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1, PSO3
CO4	Create complete NLP solutions by designing and implementing language models, sentiment classifiers, vector embeddings, sequence tagging models (using RNNs, LSTMs, GRUs), and topic models (LSA, LDA) for real-world textual data	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3

TEXT BOOK:

1. Daniel Jurafsky, James H. Martin, “An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition”, 3rd Edition, Pearson Publication, 2014.

REFERENCE BOOK:

1. Steven Bird, Ewan Klein and Edward Loper, “Natural Language Processing with Python”, 1st Edition, O_Reilly Media, 2009.

22IT812 DATA COMMUNICATIONS

Image source: https://img.brainkart.com/subject/221.jpg

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Basics of computers and Computer programming

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the principles of data communication and the basic communication protocols. The course emphasizes design issues involved in the design and utilization communication protocols. including data link layer, network layer, Transport layer, Application Protocols.

MODULE-1

UNIT-1 **12L+T+6P=18 Hours**

BASICS OF PHYSICAL LAYER AND DATA LINK LAYER:

Data and Signals, Performance metrics, Transmission modes, multiplexing and Spreading, Transmission media, switching, Error Detection and Correction, Data Link Control, Multiple Access, wired LAN, wireless LAN, Connecting Devices,

UNIT-2 **12L+0T+10P=22 Hours**

DATA LINK LAYER PROTOCOLS:

Error Detection and Correction methods, Linear Block codes, check sum, Framing, simplex, stop and wait protocol, stop and wait ARQ, Goback n ARQ protocol, selective repeat ARQ, ALOHA, CSMA, CSMA/CD, CSMA/CA.

PRACTICES:

- Hamming Distance
- Check Sum
- Framing of Binary Stream
- Implementation of simplex, stop and wait protocols
- CSMA CD
- CSMA CA

MODULE-2

UNIT-1 **12L+0T+6P=18 Hours**

NETWORK AND TRANSPORT LAYERS:

Internet working, IPv4, Address mapping, ICMP, Error Reporting, IGMP, Routing protocols congestion control and quality of service, DNS.

UNIT-2 **12L+0T+10P=22 Hours**

NETWORK AND TRANSPORT LAYERS PROTOCOLS:

Fragmentation, checksum, dynamic routing, link state routing, multi cast routing, token bucket.

PRACTICES:

- Fragmentation.
- Routing algorithms.
- Congestion control algorithms.

SKILLS:

- ✓ *Understand the working of Local Area Network and Networking Devices.*
- ✓ *Learn different Error detection and correction methods.*
- ✓ *Design a network for an office.*
- ✓ *Understanding the network layer protocols.*

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the fundamentals of physical and data link layers including transmission techniques, multiplexing, and access control mechanisms.	PO1, PO2, PO3, PO5, PSO2
CO2	Analyze various data link layer protocols such as ARQ variants, CSMA/CD, and error correction mechanisms.	PO1, PO2, PO4, PO5, PO10, PSO1, PSO2
CO3	Evaluate network layer protocols including IPv4, routing algorithms, and congestion control techniques for efficient data delivery.	PO1, PO2, PO4, PO5, PO6, PO10, PSO1, PSO2, PSO3
CO4	Create modular solutions using network and transport layer features like ICMP, checksum, fragmentation, and token bucket mechanisms.	PO1, PO3, PO5, PO11, PO12, PSO1, PSO2, PSO3

TEXT BOOK:

1 Behrouz A.Forouzan, “Data Communications and networking “, 4th Edition, McGrahill, 2011.

REFERENCE BOOKS:

1. William stallings, “Data & Computer Communications”, 8th Edition, Pearson Publishers, 2007
2. Jim kurose & Ross , “Computer Networking: A Top-Down Approach”,8th Edition, Pearson Publishers, 2021

22IT826 NETWORK PROTOCOLS

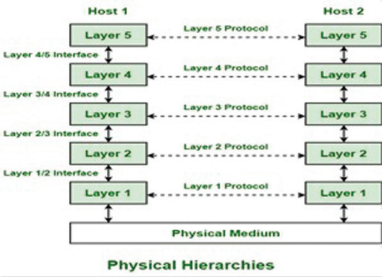


Image source: https://media.geeksforgeeks.org/wp-content/uploads/20200712173138/5615.png

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Basics of Computer Networks

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the protocols of different types of network architectures. The course emphasizes design issues involved in the design of networking protocols, including network layer protocols, Transport layer protocols and Application layer protocols.

MODULE-1

UNIT-1 6L+6T+0P=12 Hours

APPLICATION LAYER PROTOCOLS:

BOOTP, DHCP, DNS, Finger, FTP, HTTP, S-HTTP, IMAP and IMAP4, MIME (S-MIME), NAT, NNTP, POP and POP3

UNIT-2 10L+10T+0P=20 Hours

E-MAIL AND NETWORK MANAGEMENT PROTCOLS:

Rlogin, RMON, SLP, SMTP, SNMP, SNMPv1, SNMPv2, SNMPv3, SNTP, TELNET, URL, Whois (and RWhois), XMPP, X Window/X Protocol, RPC,

PRACTICES:

- DHCP
- IMAP
- POP
- Rlogin
- SMTP
- SNMP
- RPC

MODULE-2

UNIT-1 8L+8T+0P=16 Hours

LAN PROTOCOLS:

PPP: Point-to-Point Protocols, FTP: File Transfer Protocol, PAP: Password Authentication Protocol, Frame Relay, WAN Protocol for Internetworking, LAPF Link Access Procedure for Frame Mode Services, HDLC High Level Data Link Control;

IEEE 802.3 Local Area Network protocols, Fast Ethernet 100Mbps Ethernet (IEEE 802.3u), Gigabit (1000 Mbps), IEEE 802.3z (1000Base-X), and 802.3ab (1000Base-T), 10-Gigabit Ethernet: IEEE 802.3ae, and 802.3an, WLAN: Wireless LAN by IEEE 802.11 protocols, IEEE 802.11i WLAN Security Standard , IEEE 802.1X EAP over LAN(EAPOL) for LAN/WLAN Authentication and Key Management, WPAN: Wireless.

SKILLS:

- ✓ *LAN protocol designing.*
- ✓ *WAN protocol designing.*
- ✓ *Token Ring protocol designing.*

UNIT-2 **8L+8T+0P=16 Hours**

MAC PROTCOLS:

Personal Area Network Communication Protocols – IEEE 802.15.1 and the Bluetooth for WPAN Communications – FDDI: Fiber Distributed Data Interface – Token Ring: IEEE 802.5 LAN Protocol – LLC: Logic Link Control (IEEE 802.2)- SNAP: Sub-Network Access Protocol – STP: Spanning Tree Protocol (IEEE 802.1D).

PRACTICES:

- PAP
- LAPF
- Fast Ethernet
- Authentication and Key management.
- Token Ring

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply application-layer protocols (e.g., DNS, HTTP, FTP, SMTP, XMPP) in real-world data communication and network-based systems.	PO1, PO2, PO5, PO12, PSO1, PSO2
CO2	Analyze various E-mail, management, and remote access protocols to determine their roles in Internet-based communication systems.	PO1, PO2, PO4, PO5, PO6, PO12, PSO1, PSO3
CO3	Evaluate LAN and MAC protocols such as IEEE 802.3, 802.11, 802.15, and Token Ring in terms of performance, efficiency, and applicability.	PO1, PO2, PO4, PO12, PSO2, PSO3
CO4	Create secured and interoperable communication setups using authentication, NAT, S-MIME, and EAP-based security protocols.	PO1, PO3, PO5, PO12, PSO2, PSO3

TEXT BOOK:

1. Jielin Dong, “Network Protocols Handbook”, 4th Edition, Javvin Press, 2007.

REFERENCE BOOKS:

1. Bruce Hartpence, “Packet Guide to Core Network Protocols”, OReilly Media, Inc., 2011.
2. Walter Y. Chen, “Home Network Basis: Transmission Environments and Wired/Wireless Protocols”, Pearson, 2003.
3. Ana Minaburo , Laurent Toutain, “Local Networks and the Internet: From Protocols to Interconnection”, John Wiley and Sons, 2011.

22IT831 WIRELESS NETWORKS

Image source: https://www.shutterstock.com/image-photo/smart-agriculture-vertical-farm-sensor-technology-772107739 https://www.shutterstock.com/image-illustration/vehicles-infrared-sensor-device-unmanned-smart-781805650

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Computers networks, Basics of sensor

COURSE DESCRIPTION AND OBJECTIVES:

It covers the fundamentals of wireless sensor design and radio frequency (RF) technology, the communication protocols used, and the application requirements of this technology. Further, it covers Mobile Adhoc Networks and to study about localization and routing protocols. And also covers the concepts of data management and to acquire the knowledge of operating systems and infrastructure establishment in wireless networks.

MODULE-1

UNIT-1 **8L+8T+0P=16 Hours**

INTRODUCTION TO SENSOR NETWORKS:

Introduction to Sensor Networks, unique constraints and challenges in WSN, Advantage of Sensor Networks, Applications of Sensor Networks, Mobile Adhoc NETworks (MANETs) and Wireless Sensor Networks, Design principles for WSNs.

UNIT-2 **8L+8T+0P=16 Hours**

LOCALIZATION AND ROUTING:

Localization and positioning, Coverage and connectivity, Single-hop and multihop localization, self-configuring localization systems, sensor management Network Protocols: Issues in designing MAC protocol for WSNs, Classification of MAC Protocols, S-MACProtocol, B-MAC protocol, IEEE 802.15.4 standard and Zig Bee, Dissemination protocol for large sensor network. Routing protocols: Issues in designing routing protocols, Classification of routing protocols, Energy-efficient routing, Unicast, Broadcast and multicast, Geographic routing.

PRACTICES:

- Understand Types of sensors.
- Know Network Defenses.
- Place your sensor Devices Correctly.
- Don’t Disable Personal Firewalls.
- Use Centralized Logging and Immediate Log Analysis.

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

DATA MANAGEMENT AND MANIPULATION:

Storage and retrieval in network, compression technologies for WSN, Data aggregation technique, Data management in WSN, Storage and indexing in sensor networks, Query processing in sensor, Directed Diffusion, Tiny aggregation, greedy aggregation, security in WSN.

Operating Systems for Wireless Networks: Introduction - Operating System Design Issues - Examples of Operating Systems – TinyOS – Mate – MagnetOS – MANTIS - OSPM - EYES OS – SenOS – EMERALDS – PicOS

SKILLS:

- ✓ *Embedded Systems (for sensor nodes).*
- ✓ *Power Saving Techniques (long term, short term, solar?).*
- ✓ *Delay-Tolerant Networking (harsh environments).*
- ✓ *Wireless Coverage (line of sight? mesh network? etc).*

UNIT-2 **8L+8T+0P=16 Hours**

INFRASTRUCTURE ESTABLISHMENT:

Topology Control, controlling topology in flat networks, Hierarchical networks, Clustering, Time Synchronization, Sensor Tasking and control

PRACTICES:

- Use strong password protection and authentication.
- Connect to secure Wi-Fi.
- Enable firewall protection at work and at home.
- Invest in sensor systems.
- Institutional reform, strategic planning.
- Legal and regulatory frameworks, unbundling and competition.
- Contractual arrangements, sources of financing the allocation of risk.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply design principles, architectures, and protocols for Wireless Sensor Networks in various application domains.	PO1, PO2, PO3, PO5, PSO1, PSO3
CO2	Analyze localization, routing, and MAC protocols in sensor networks for energy efficiency and scalability.	PO1, PO2, PO4, PO5, PSO1, PSO2
CO3	Evaluate storage, indexing, and aggregation strategies along with WSN-specific operating systems.	PO2, PO4, PO6, PO12, PSO1, PSO2
CO4	Create energy-aware and time-synchronized network infrastructure using clustering, tasking, and control mechanisms.	PO1, PO3, PO5, PO11, PO12, PSO1, PSO2, PSO3

TEXT BOOK:

1. HolgerKerl, Andreas Willig, “Protocols and Architectures for Wireless Sensor Network”, JohnWiley and Sons, 2005.

REFERENCE BOOKS:

1. Raghavendra, Cauligi S, Sivalingam, Krishna M., ZantiTaieb, “Wireless Sensor Network”, 1st Edition, Springer, 2004.
2. Feng Zhao, Leonidas Guibas, “ Wireless Sensor Network”, 1st Edition, Elsevier, 2004.
3. Kazem, Sohraby, Daniel Minoli, TaiebZanti, “Wireless Sensor Network: Technology, Protocolsand Application”, 1st Edition, John Wiley and Sons, 2007.

22IT811 CYBER SECURITY

https://www.google.com/search?q=cyber+security&client

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Information Security

COURSE DESCRIPTION AND OBJECTIVES:

Protect and defend computer systems and networks from cybersecurity attacks. Diagnose and investigate cybersecurity events or crimes related to computer systems and digital evidence. Effectively communicate in a professional setting to address information security issues.

MODULE-1

UNIT-1 **6L+6T+0P=12 Hours**

INTRODUCTION TO CYBER SECURITY:

Introduction -Computer Security - Threats -Harm - Vulnerabilities - Controls - Authentication - Access Control and Cryptography - Web—User Side - Browser Attacks - Web Attacks.

UNIT-2 **10L+10T+0P=20 Hours**

SECURITY IN OPERATING SYSTEM:

Network Security in Operating Systems - Security in the Design of Operating Systems -Rootkit -Network security attack- Threats to Network Communications -Wireless Network Security - Denial of Service - Distributed Denial-of-Service.

PRACTICES:

- protect your personal computer system by creating User Accounts with Passwords and types of User Accounts for safety and security.
- Practice a password protect for the word document.
- Authentication practices.
- Browser attacks.
- web attacks.

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

CRYPTOGRAPHY IN NETWORK SECURITY:

Firewalls - Intrusion Detection and Prevention Systems - Network Management - Databases - Security Requirements of Databases -Reliability and Integrity - Database Disclosure - Data Mining and Big Data.

UNIT-2 **8L+8T+0P=16 Hours**

PRIVACY IN CYBERSPACE:

Privacy Concepts -Privacy Principles and Policies -Authentication and Privacy - DataMining -Privacy on the Web - Email Security - Privacy Impacts of Emerging Technologies- Where the Field Is Headed.

SKILLS:

- ✓ *Networking and System Administration.*
- ✓ *Knowledge of Operating Systems and Virtual Machines.*
- ✓ *Network Security Control.*

PRACTICES:

- Firewall and network testing tools.
- Fire wall configuration.
- Database Reliability and Integrity.
- Impacts of emerging technologies.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply foundational principles of cybersecurity to detect threats, vulnerabilities, and controls in computing environments.	PO1, PO2, PO6, PSO2
CO2	Analyze operating system and network-level security threats, including DoS, DDoS, and rootkits, to propose suitable countermeasures.	PO2, PO3, PO4, PO5, PO8, PO10, PSO1, PSO2
CO3	Evaluate the effectiveness of cryptographic algorithms, firewalls, IDS/IPS, and database security techniques in preserving integrity.	PO2, PO4, PO6, PO12, PSO1, PSO2
CO4	Create policies and strategies for ensuring privacy in cyberspace, including privacy-aware authentication and secure web practices.	PO3, PO5, PO6, PO11, PO12, PSO2, PSO3

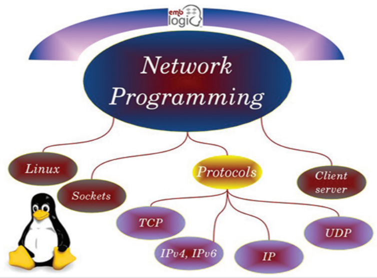
TEXT BOOK:

1. Charles P. Pfleeger Shari Lawrence Pfleeger Jonathan Margulies, “Security in Computing”, 5th Edition, Pearson Education, 2015

REFERENCE BOOKS:

1. George K. Kostopoulous, “Cyber Space and Cyber Security”, CRC Press, 2013.
2. MarttiL ehto, Pekka Neittaanmäki, “Cyber Security: Analytics, Technology and Automation”, edited, Springer International Publishing Switzerland, 2015.
3. Nelson Phillips and Enfinger Steuart, “Computer Forensics and Investigations”, Cengage Learning, 2009.

22IT825 NETWORK PROGRAMMING



https://www.google.com/ search?q=network+programming +&tbm=isch& ved= 2ahUKEwjJvfPE__f4AhVHzqACHQiuAqUQ2-cCegQIABAA&oq=network+programming+&gs_lcp

Hours Per Week :

L	T	P	C
3	-	2	4

PREREQUISITE KNOWLEDGE: Computer networks and Computer programming

COURSE DESCRIPTION AND OBJECTIVES:

This course will cover the practical aspects of computer network programming, with emphasis on the Internet. The goal of this course is to introduce the students to the computer networks, Internet programming, TCP/IP protocol stack and some of its important protocols.

MODULE-1

UNIT-1 **14L+0T+6P=20 Hours**

NETWORKING & TCP/IP:

Communication protocols, Network architecture, UUCP, XNS, IPX/SPX for LANs, TCP & IP headers, IPv4 & v6 address structures, Programming Applications: Time & date routines, Internet protocols: Application layer, Transport layer, Network layer, Datalink layer protocols, Chat, Email, Web server working method & programming.

UNIT-2 **10L+0T+10P=20 Hours**

SOCKET PROGRAMMING:

Creating sockets, Posix data type, Socket addresses, Assigning address to a socket, Java socket programming, Thread programming, Berkeley Sockets: Overview, socket address structures, byte manipulation & address conversion functions, elementary socket system calls – socket, connect, bind, listen, accept, fork, exec, close, TCP ports (ephemeral, reserved), Berkeley Sockets: I/O asynchronous & multiplexing models, select & poll functions, signal & fcntl functions, socket implementation (client & server programs), UNIX domain protocols.

PRACTICES:

- Client and iterative server using TCP.
- Client and concurrent server using TCP.
- Client and concurrent server using UDP.
- Retrieve date and time using TCP.
- Retrieve date and time using UDP.

MODULE-2

UNIT-1 **12L+0T+6P=18 Hours**

APIS & WINSOCK PROGRAMMING:

APIs & Winsock Programming: Windows socket API, window socket & blocking I/O model, blocking sockets, blocking functions, timeouts for blocking I/O, API overview, Different APIs & their programming technique, DLL & new API’s, DLL issues, Java Beans.

UNIT-2 **12L+0T+10P=22 Hours**

WEB PROGRAMMING & SECURITY:

Java network programming, packages, RMI, Overview of Java script, WAP architecture & WAP services, Web databases, Component technology, CORBA concept, CORBA architecture, CGI programming, Firewall & security technique, Cryptography, Digital Signature.

SKILLS:

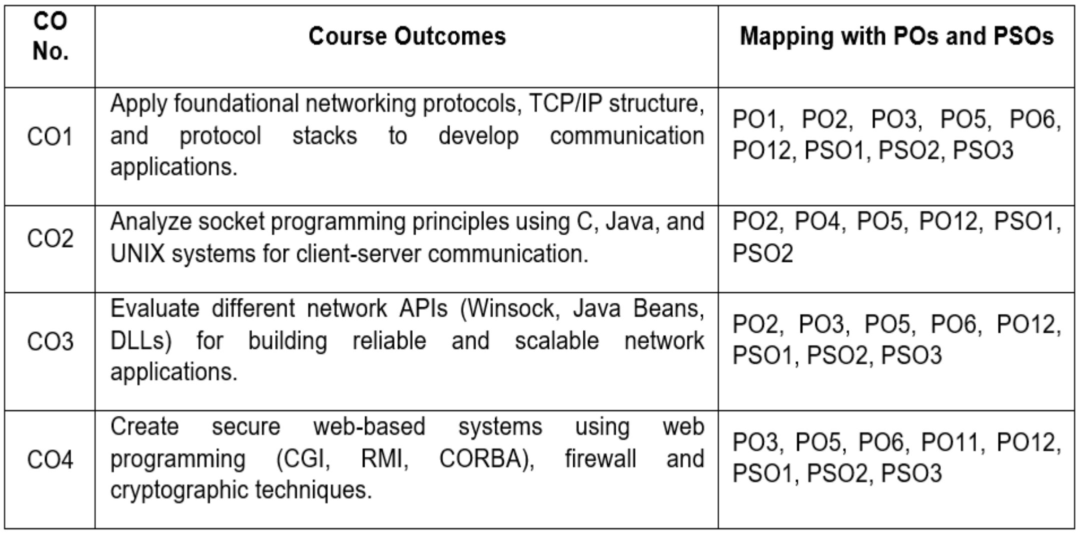
- ✓ *Threads for developing high performance scalable applications.*
- ✓ *Socket programming using TCP Sockets, UDP sockets.*

PRACTICES:

- Window socket, blocking sockets.
- Blocking functions.
- Client and server routines showing Blocking I/O.
- Client and server routines showing I/O multiplexing.
- CORBA architecture.
- Unix domain stream socket.
- Unix domain Datagram socket.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply foundational networking protocols, TCP/IP structure, and protocol stacks to develop communication applications.	PO1, PO2, PO3, PO5, PO6, PO12, PSO1, PSO2, PSO3
CO2	Analyze socket programming principles using C, Java, and UNIX systems for client-server communication.	PO2, PO4, PO5, PO12, PSO1, PSO2
CO3	Evaluate different network APIs (Winsock, Java Beans, DLLs) for building reliable and scalable network applications.	PO2, PO3, PO5, PO6, PO12, PSO1, PSO2, PSO3
CO4	Create secure web-based systems using web programming (CGI, RMI, CORBA), firewall and cryptographic techniques.	PO3, PO5, PO6, PO11, PO12, PSO1, PSO2, PSO3

TEXT BOOKS:

1. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, “UNIX Network Programming”, 4th Edition, Pearson Education, 2015.
2. W. Richard Stevens, “UNIX Network Programming”, 1st Edition, PHI, 2009.

REFERENCE BOOKS:

1. Graham Glass, King abls, “UNIX for Programmers and Users”, 3rd Edition, Pearson Education, 2018.
2. M. J. Rochkind, “Advanced UNIX Programming”, 2nd Edition, Pearson Education, 2008.

22IT833 BLOCKCHAIN TECHNOLOGIES

Image source: https://g.foolcdn.com/editorial/images/468747/blockchain-cryptocurrency-bitcoin-ethereum-litecoin-ripple-getty.jpg

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Network system, database storage, cryptography

COURSE DESCRIPTION AND OBJECTIVES:

This course builds understand how block chain systems (mainly Bit coin and Ethereum) work and to securely interact with them. Design and deploy smart contracts and distributed applications. Integrate ideas from block chain technology into their own projects. Demonstrate the foundation of the Block chain technology and understand the processes in payment and funding. Identify the risks involved in building Block chain applications. Review of legal implications using smart contracts. Choose the present landscape of Blockchain implementations and Understand Crypto currency markets. Examine how to profit from trading crypto currencies.

MODULE-1

UNIT-1 **12L+8T+0P=20Hours**

EVOLUTION OF BLOCKCHAIN AND BLOCKCHAIN CONCEPTS:

Introduction, Scenarios, Challenges Articulated, Blockchain, Blockchain Characteristics, Opportunities Using Blockchain, History of Blockchain.

Evolution of Blockchain: Evolution of Computer Applications, Centralized Applications, Decentralized Applications, Stages in Blockchain Evolution, Consortia, Forks, Public Blockchain Environments, Type of Players in Blockchain Ecosystem, Players in Market.

UNIT-2 **12L+8T+0P=20Hours**

BLOCKCHAIN CONCEPTS:

Blockchain Concepts: Introduction, Changing of Blocks, Hashing, Merkle-Tree, Consensus, Mining and Finalizing Blocks, Currency aka tokens, security on blockchain, data storage on blockchain, wallets, coding on blockchain: smart contracts, peer-to-peer network, types of blockchain nodes, risk associated with blockchain solutions, life cycle of blockchain transaction.

PRACTICES:

- Discuss Blockchain Characteristics.
- Connectivity of Centralized and Decentralized network Applications.
- How to create Public Blockchain Environments.
- Discus security on blockchain.
- Discus data storage on blockchain.
- Explain life cycle of blockchain transaction.

MODULE-2

UNIT-1 **12L+8T+0P=20Hours**

ARCHITECTING BLOCKCHAIN SOLUTIONS AND ETHEREUM BLOCKCHAIN IMPLEMENTATION:

Architecting Blockchain solutions: Introduction, Obstacles for Use of Blockchain, Blockchain Relevance Evaluation Framework, Blockchain Solutions Reference Architecture, Types of Blockchain

SKILLS:

- ✓ *To develop Public Blockchain Environments.*
- ✓ *Develop source code for Blockchain applications.*
- ✓ *To analysis peer-to-peer network for Blockchain applications.*
- ✓ *To develop architecture for Blockchain Platforms*

Applications. Cryptographic Tokens, Typical Solution Architecture for Enterprise Use Cases, Types of Blockchain Solutions, Architecture Considerations, Architecture with Blockchain Platforms, Approach for Designing Blockchain Applications.

UNIT-2 **12L+8T+0P=20Hours**

ETHEREUM BLOCKCHAIN IMPLEMENTATION AND ADVANCED CONCEPTS IN BLOCK CHAIN:

Ethereum Blockchain Implementation: Introduction, Tuna Fish Tracking Use Case, Ethereum Ecosystem, Ethereum Development, Ethereum Tool Stack, Ethereum Virtual Machine, Smart Contract Programming, Integrated Development Environment, Truffle Framework, Ganache, Unit Testing, Ethereum Accounts, MyEtherWallet, Ethereum Networks/Environments, OpenZeppelin Contracts Hyperledger Blockchain Implementation, Introduction, Use Case – Car Ownership Tracking, Hyperledger Fabric, Hyperledger Fabric Transaction Flow, FabCar Use Case Implementation, Invoking Chaincode Functions Using Client Application.

Advanced Concepts in Blockchain: Introduction, InterPlanetary File System (IPFS), Zero-Knowledge Proofs, Oracles, Self-Sovereign Identity, Blockchain with IoT and AI/ML Quantum Computing and Blockchain, Initial Coin Offering, Blockchain Cloud Offerings, Blockchain and its Future Potential.

PRACTICES:

- Explain Blockchain Relevance Evaluation Framework.
- What is Cryptographic Tokens.
- Discus Architecture with Blockchain Platforms.
- Discus on Ethereum Development, Ethereum Tool Stack, Ethereum Virtual Machine.
- Discus on OpenZeppelin Contracts Hyperledger Blockchain Implementation.
- Explain Hyperledger Fabric Transaction Flow.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the foundational concepts of blockchain, including decentralization, hashing, Merkle trees, and consensus, to understand how modern blockchain systems ensure trust and security	PO1, PO2, PO3, PO5, PO9, PO10, PO12, PSO1, PSO2, PSO3
CO2	Analyze the evolution of blockchain technology from centralized to decentralized architectures, evaluate blockchain types (public, private, consortia), and assess their roles in real-world applications.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12, PSO1, PSO2, PSO3
CO3	Evaluate core blockchain mechanisms such as mining, finalizing blocks, tokenization, and wallet usage for their effectiveness in maintaining distributed ledger integrity and network consensus.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12, PSO1, PSO2, PSO3
CO4	Create simple smart contracts and demonstrate secure peer-to-peer interaction by simulating blockchain applications involving data storage, token transactions, and validation processes	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12, PSO1, PSO2, PSO3

TEXT BOOKS:

1. Ambadas, Arshad Sarfarz Ariff, Sham “Blockchain for Enterprise Application Developers”, Wiley, 2020.
2. Andreas M. Antonpoulos, “Mastering Bitcoin: Programming the Open Blockchain”, O’Reilly, 2021.

REFERENCE BOOKS:

1. Joseph Bambara, Paul R. Allen, “Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions, Mc Graw Hill, 2018.

22IT809 COMPUTER ORGANIZATION AND ARCHITETCURE

SOURCE: https://www.siteforinfotech.com/wp-content/uploads/2013/07/Solved-MCQ-of-Computer-Organization-and-Architecture.jpg

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Basics of Computers, Digital Electronics.

COURSE DESCRIPTION AND OBJECTIVES:

This course covers the basics of modern Computer Organization and Architecture. The emphasis is on understanding the design of computer and its components. The student will learn the concepts of data representation, micro operations, memory organizations and input output organization. Case study of 8086 helps the students to visualize the basic concepts of the course.

MODULE-1

UNIT-1 **6L+6T+0P=12 Hours**

RTL, DATA REPRESENTATIONS AND BASIC CPU ORGANIZATION:

Introduction: Organization and Architecture; Structure and function; Register transfer; Bus and memory transfer.

Data Representations and Micro Operations: Fixed point representation and Floating-point representation; Overflow; Micro operations.

Basic CPU Organization: Instruction codes; Computer registers; Computer instructions; Instruction formats; Addressing modes; Stack organization.

UNIT-2 **10L+10T+0P=20 Hours**

DIGITAL CIRCUITS AND IMPLEMENTING MICRO OPERATIONS:

Designing Digital Circuits: Designing binary adder; Binary adder-subtractor; Binary incrementer; Arithmetic circuit; Logic circuit; Shift circuit; Arithmetic logic shift unit.

Implementing Micro Operations: Instruction cycle; Memory reference instructions; Register reference instructions, Input output instructions; Interrupt cycle.

PRACTICES:

- Arithmetic operations such as addition, subtraction etc., on binary data and determining the occurrence of overflow in addition.
- Designing digital circuits required to perform various operations.
- Understanding CPU Organization for different configurations of computer system.

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

INTRODUCTION TO MEMORY, PERIPHERAL DEVICES AND PARALLEL PROCESSING:

Memory Introduction: Memory hierarchy; Main memory, Associative memory, Cache memory, Virtual memory.

Peripherals Introduction: Peripheral devices; Asynchronous data transfer.

Parallel Processing Introduction: Parallel Processing; Pipelining concepts.

SKILLS:

- ✓ *Students are able to develop different digital circuits required to perform the micro operations.*
- ✓ *Students will identify different pipeline hazards and able to classify the organization and architecture of computer systems.*
- ✓ *Students are able to analyze the performance of memory, I/O and pipelined systems.*
- ✓ *Students will interface memory and peripheral with the central system*

UNIT-2 **8L+8T+0P=16 Hours**

MEMORY & I/O INTERFACING, PIPELINE IMPLEMANTATIONS:

Memory Interfacing: Main memory interface with CPU; Address mapping in cache memory and Virtual memory.

I/O interfacing: Input-output interface; Modes of transfer; Priority interrupt; Direct memory access.

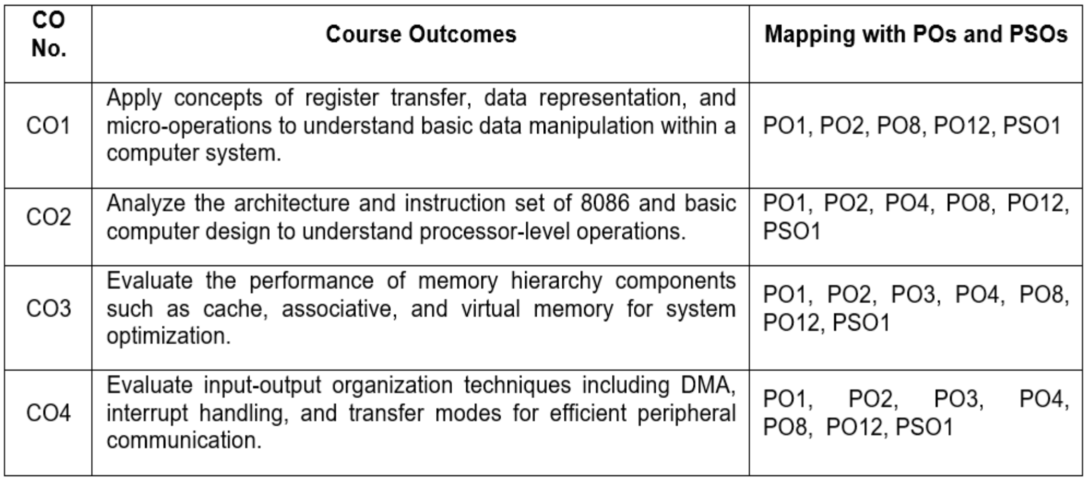
Pipeline Implementations: Arithmetic Pipeline; Instruction Pipeline; RISC Pipeline.

PRACTICES:

- Analyzing the performance of pipelined systems
- Address mapping in cache and virtual memory, interfacing memory with CPU.
- ASCII Codes for various characters, Interfacing input and output devices with central system.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply concepts of register transfer, data representation, and micro-operations to understand basic data manipulation within a computer system.	PO1, PO2, PO8, PO12, PSO1
CO2	Analyze the architecture and instruction set of 8086 and basic computer design to understand processor-level operations.	PO1, PO2, PO4, PO8, PO12, PSO1
CO3	Evaluate the performance of memory hierarchy components such as cache, associative, and virtual memory for system optimization.	PO1, PO2, PO3, PO4, PO8, PO12, PSO1
CO4	Evaluate input-output organization techniques including DMA, interrupt handling, and transfer modes for efficient peripheral communication.	PO1, PO2, PO3, PO4, PO8, PO12, PSO1

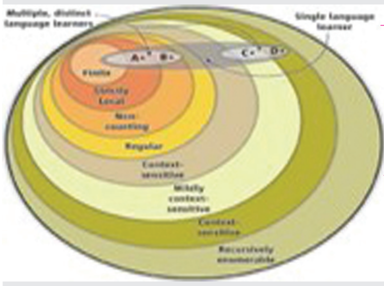
TEXT BOOK:

1. M.Moris Mano, “Computer Systems Architecture”, 3rd edition, Pearson/Prentice Hall India, 2007.

REFERENCE BOOKS:

1. William Stallings, “Computer Organization and Architecture”, 6th Edition, Pearson/Prentice Hall India, 2007.
2. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, 5th Edition, Tata McGraw Hill, 2007.
3. David A Patterso and John L Hennessy, “Computer Organization and Design - The Hardware/ Software Interface, ARM edition”, 5th Edition, Elsevier, 2009.

22IT817 FORMAL LANGUAGES AND AUTOMATA THEORY



Source: https://images.app.goo.gl/vscRhXSi5upvmoyt5

Hours Per Week :

L	T	P	C
3	2	-	4

PREREQUISITE KNOWLEDGE: Computer knowledge, Discrete mathematics

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the basic theory of Computer Science and formal methods of computation like automata theory, formal languages, grammars and Turing Machines. The objective of this course is to explore the theoretical foundations of computer science from the perspective of formal languages and classify machines by their power to recognize languages.

MODULE-1

UNIT-1 **12L+8T +0P=20 Hours**

INTRODUCTION TO AUTOMATA:

Alphabets, Strings and languages, Automata and Grammars, Regular languages, Deterministic finite automata (DFA)-Formal definition, Simplified notation, State transition graph, Transition table, Language of DFA; Nondeterministic finite automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of finite automata.

UNIT-2 **12L+8T +0P=20 Hours**

REGULAR EXPRESSION:

Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleen’s Theorem, Regular expression to FA,Kleen’s Theorem, Regular expression to FA, DFA to regular expression, Arden theorem, Non regular languages, Pumping lemma for regular languages, Application of pumping lemma, Closure properties of regular languages.

PRACTICES:

- Draw a DFA for the language accepting strings ending with ‘abb’ over input alphabets
- $\sum = \{a, b\}$
- Draw a DFA for the language accepting strings starting with ‘ab’ over input alphabets
- $\sum = \{a, b\}$
- Check if the Language L = {w $\in$ {0, 1}* : w is the binary representation of a prime number} is a regular or non-regular language.
- Prove that the Language L = {1^n : n is a prime number} is a non-regular Language.
- Prove that the language L = = {0^n1^n : n >= 0} is not a regular language.

MODULE-2

UNIT-1 **12L+8T +0P=20 Hours**

GRAMMAR FORMALISM:

Regular Grammars-Right linear and left linear grammars, Equivalence between regular linear grammar and FA; Context free grammar, Definition, Examples, Derivation, Derivation trees.

Push Down Automata (PDA): Description and definition, Instantaneous description, Language of PDA, Acceptance by final state, Acceptance by empty stack, Deterministic PDA,

SKILLS:

- ✓ *Design automata, regular expressions and context free grammars for accepting or generating a certain language.*
- ✓ *Describe the language accepted by an automata or generated by a regular expression or a context free grammar.*
- ✓ *Transform between equivalent deterministic and non-deterministic finite automata, and regular expressions.*
- ✓ *Minimize finite automata and grammars of context free languages.*

UNIT-2 **12L+8T +0P=20 Hours**

TURING MACHINES (TM):

Basic model, Definition and representation, Instantaneous Description, Language acceptance by TM, Computable functions, Types of Turing machines, universal TM, Church's Thesis, Recursive and recursively enumerable languages.

PRACTICES:

- Provide an algorithm for converting a left linear grammar to a right linear grammar.
- Give pushdown automata that recognize the following languages
  - o $A = \{ w \in \{0, 1\} * \mid w$ contains at least three 1^s $\}$
  - o $B = \{ w \in \{0, 1\} * \mid w = w^R$ and the length of w is odd $\}$
- Consider the language $L = \{ w \in \Sigma^* \mid w$ is a string of balanced digits $\}$ over $\Sigma = \{ 0, 1 \}$
- Design Turing Machine for the language $L = \{0^{2\,n} : n \geq 0\}$
- Design Turing machines for the following language, The set of strings with an equal number of 0's and 1's.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the basic properties of formal languages and grammars.	PO1, PO2, PO3, PO9, PO10, PO12, PSO1
CO2	Analyze to construct automata, regular expression for any pattern.	PO1, PO2, PO3, PO9, PO10, PO12, PSO1
CO3	Evaluate the concepts relating to the theory of computation and computational models including decidability and intractability.	PO1, PO2, PO3, PO9, PO10, PO12, PSO1
CO4	Design Turing machines for any language.	PO1, PO2, PO3, PO9, PO10, PO12, PSO1

TEXT BOOKS:

1. Hopcroft and Ullman, "Introduction to Automata Theory, Languages and Computation", 2nd Edition, Pearson/Prentice Hall India, 2007.
2. Peter Linz, "An Introduction to Formal Language and Automata", Narosa Pub. House, 2011.

REFERENCE BOOKS:

1. Martin J. C., "Introduction to Languages and Theory of Computations", 2nd Edition, TataMcGraw Hill, 2005.
2. Papadimitrou, C. and Lewis, C.L., "Elements of the Theory of Computation", 2nd Edition, Pearson/Prentice Hall India, 2009.

22IT807 COMPILER DESIGN

High Level Language
Compiler
Low Level Language
Compilation Error

Source : https://images.app.goo.gl/Xz3qjyMUYEExDrVA8

Hours Per Week :

L	T	P	C
3	2	-	4

PREREQUISITE KNOWLEDGE: Computer knowledge, Discrete mathematics

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the concepts of lexical analyzer, parser, code generation and code optimization techniques. The objective of this course is to enable the student to acquire the knowledge of various phases of compiler such as lexical analyzer, parser and code optimization.

MODULE-1

UNIT-1 **12L+8T +0P=20 Hours**

INTRODUCTION TO COMPILING:

Compilers, Analysis of the source program, Phases of a compiler, Cousins of the Compiler, Grouping of Phases, Compiler construction tools, Lexical Analysis, Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Data structures in compilation, LEX, Lexical analyzer generator.

UNIT - 2 **12L+8T +0P=20 Hours**

SYNTAX ANALYSIS:

Role of the parser, Writing Grammars, Context-Free Grammars; Top-Down Parsing-Recursive Descent Parsing, Predictive Parsing; Bottom-Up Parsing-Shift Reduce Parsing, Operator Precedent Parsing; LR Parsers–SLR Parser, Canonical LR Parser, LALR Parser; YACC – automatic parser generator.

PRACTICES:

- Construct the target code for the following statement : while (x < a+b) x = 2*x;
- Show the compiler output and the interpreter output for the following Java source code:
  for (i=1; i<=4; i++) System.out.println (i*3);
- Explain the various phases of a compiler in detail. Also write down the output for the following expression: position: =initial + rate * 60
- Eliminate immediate left recursion for the following grammar :
  E->E+T | T   T->T* F | F   F-> (E) | id
- Compare and contrast LR and LL Parsers.

MODULE-2

UNIT-1 **12L+8T +0P=20 Hours**

INTERMEDIATE CODE GENERATION:

Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking.

UNIT-2 **12L+8T +0P=20 Hours**

CODE GENERATION AND CODE OPTIMIZATION:

Issues in Code Generation - Design of a simple Code Generator, Principal Sources of Optimization, Peep-hole optimization - Basic blocks, Flow graphs, DAG- Optimization of Basic Blocks-Global Data Flow Analysis.

SKILLS:

- ✓ *Design various parsers using top-down and bottom-up approaches.*
- ✓ *Usage of generators like LEX and YACC.*
- ✓ *Analyze recognizer for programming language.*

PRACTICES:

- What is a three address code? Mention its types. How would you implement the three address statements? Explain with examples.
- Construct a quadruple, triples for the following expression: a +a*(b-c)+(b-c)*d?
- Illustrate loop optimization with suitable example.
- Explain various method to handle peephole optimization.
- Implementation of the Principal sources of optimization techniques.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the knowledge of compiler phases to analyze and design lexical analyzers using tools like LEX.	PO1, PO2, PO3, PO9, PO12, PSO1
CO2	Analyze grammar rules and parsing techniques to develop syntactic analyzers using top-down and bottom-up parsing, including YACC.	PO1, PO2, PO3, PO9, PO12, PSO1
CO3	Evaluate syntax-directed definitions and intermediate representations (TAC, DAG) for effective semantic analysis and code translation.	PO1, PO2, PO3, PO9, PO12, PSO1
CO4	Create optimized target code using basic blocks, flow graphs, peep-hole optimization, and global data flow analysis.	PO1, PO2, PO3, PO9, PO12, PSO1

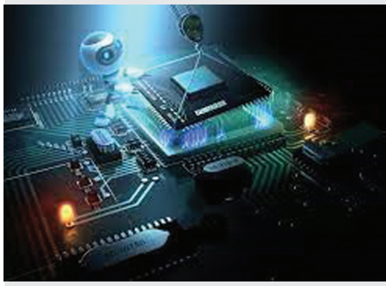
TEXT BOOKS:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques and Tools, 2nd Edition, Pearson Education, 2009.
2. Andrew W. Appel, Jens Palsberg Modern compiler implementation in Java, ‖ 2nd Edition, Cambridge University Press, 2002.

REFERENCE BOOKS:

1. Allen I. Holub, “Compiler Design in C”, 1st edition, Prentice Hall of India, 2003.
2. C. N. Fischer and R. J. LeBlanc, “Crafting a compiler with C”, 1st Edition, Benjamin Cummings, 2003.
3. J.P. Bennet, “Introduction to Compiler Techniques”, 2nd Edition, Tata McGraw-Hill, 2003.

22IT801 ADVANCED COMPUTER ARCHITECTURE



SOURCE: https://wallpaperaccess.com/full/1433353.jpg

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Computer organization and architecture

COURSE DESCRIPTION AND OBJECTIVES:

This course covers the basics of evolution of computer architecture. The emphasis is on understanding the parallel processing and inter-connection networks of multi-processor systems. The student will learn the concepts of pipelining, shared memory architectures vs message passing architectures and also about the memory hierarchy design.

MODULE-1

UNIT-1 **8L+8T+0P=16 Hours**

INTRODUCTION TO ADVANCED COMPUTER ARCHITECTURE AND PARALLEL PROCESSING:

Four Decades of Computing; Flynn's Taxonomy of Computer Architecture; SIMD Architecture; MIMD Architecture; Interconnection Networks; Review of Pipelining; Examples of some pipeline in modern processors; pipeline hazards; data hazards; control hazards; Techniques to handle hazards; CISC vs RISC.

UNIT-2 **8L+8T+0P=16 Hours**

MEMORY HIERARCHY DESIGN:

Introduction; Ten Advanced Optimizations of Cache Performance; Memory Technology and Optimizations; Protection: Virtual Memory and Virtual Machines; Crosscutting Issues: The Design of Memory Hierarchies.

PRACTICES:

- Understanding the cache optimization techniques.
- Memory Hierarchy Design case studies:
  - o ARM Cortex-A8 and
  - o Intel Core i7

MODULE-2

UNIT-1 **6L+6T+0P=12Hours**

MULTIPROCESSORS INTERCONNECTION NETWORKS:

Interconnection Networks Taxonomy; Bus-Based Dynamic Interconnection Networks; Switch-Based Interconnection Networks; Static Interconnection Networks.

UNIT-2 **10L+10T+0P=20 Hours**

SHARED MEMORY ARCHITECTURE AND MESSAGE PASSING ARCHITECTURE:

Classification of Shared Memory Systems; Bus-Based Symmetric Multiprocessors; Basic Cache Coherency Methods; Snooping Protocols; Directory Based Protocols; Introduction to Message Passing; Routing in Message Passing Networks; Switching Mechanisms in Message Passing; Processor Support for Message Passing; Example Message Passing Architectures; Message Passing Versus Shared Memory Architectures.

SKILLS:

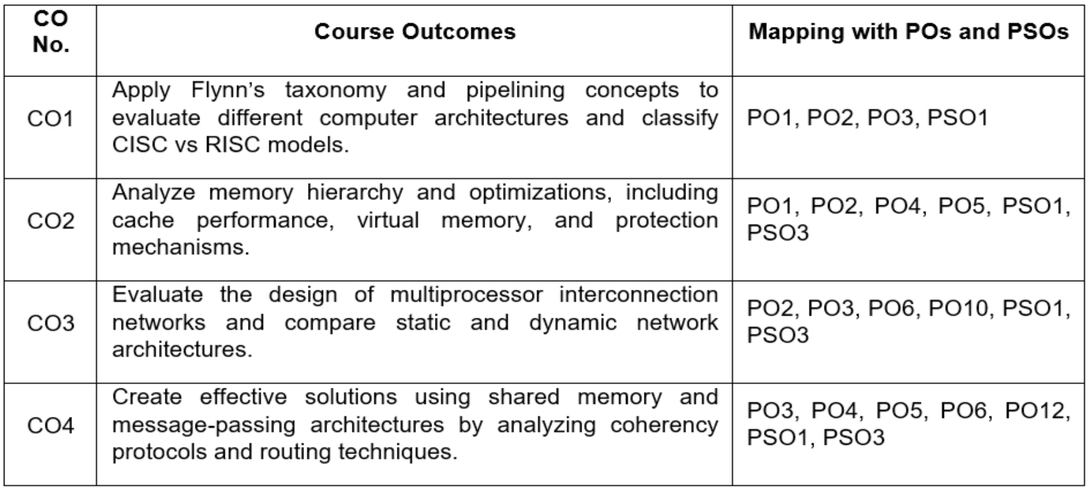
- ✓ *Students are able to analyze the performance issues in parallel processing systems.*
- ✓ *Students will identify different pipeline hazards and able to classify different architectures of computer systems*
- ✓ *Students are able to analyze the performance of shared memory and message passing systems.*
- ✓ *Design memory hierarchy for different architectures.*

PRACTICES:

- Analyzing the performance of shared memory architectures.
- Analyzing the performance of message passing architectures.
- Understanding different inter connection networks.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply Flynn's taxonomy and pipelining concepts to evaluate different computer architectures and classify CISC vs RISC models.	PO1, PO2, PO3, PSO1
CO2	Analyze memory hierarchy and optimizations, including cache performance, virtual memory, and protection mechanisms.	PO1, PO2, PO4, PO5, PSO1, PSO3
CO3	Evaluate the design of multiprocessor interconnection networks and compare static and dynamic network architectures.	PO2, PO3, PO6, PO10, PSO1, PSO3
CO4	Create effective solutions using shared memory and message-passing architectures by analyzing coherency protocols and routing techniques.	PO3, PO4, PO5, PO6, PO12, PSO1, PSO3

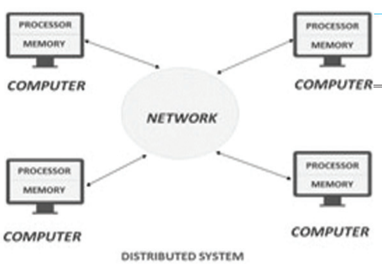
TEXT BOOKS:

1. David A. Patterson and John L. Hennessey, “Computer organization and design”, 5th Edition, Elsevier, 2014.
2. Hesham El-Rewini and Mostafa Abd-El-Barr, “Advanced Computer Architecture and Parallel Processing”, 1st Edition, A John Wiley & Sons, Inc Publication, 2005.

REFERENCE BOOK:

1. M. Moris Mano, “Computer Systems Architecture”, 3rd Edition, Pearson/Prentice Hall India, 2007.

22IT816 DISTRIBUTED SYSTEMS



Source of Image: https://i.pinimg.com/736x/e7/ee/b2/e7eeb2cf7457137362e2a478606a7788.jpg

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Basics of Computing and Operating System.

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces broad range of topics related to parallel and distributed computing, including parallel and distributed architectures and systems, parallel and distributed programming paradigms, parallel algorithms and scientific and other applications of parallel and distributed computing.

MODULE-1

UNIT-1 **8L+8T+0P=16 Hours**

DISTRIBUTED SYSTEMS AND COMMUNICATION:

Distributed Systems: Introduction, Definition, Evolution Goals of distributed systems. Communication: Message Passing, Features and Issues, Synchronization, Buffering, Process Addressing, Failure Handling, Remote procedure call (RPC): Model, Implementation, Stub generation, RPC messages, Marshalling, server Management, Call semantics, communication protocols for RPC-Client server binding, RMI.

UNIT-2 **8L+8T+0P=16 Hours**

DISTRIBUTED MODELS AND PROTOCOLS:

Distributed System Models: System models, Issues in the design of distributed systems, Distributed computing environment. Communication: Synchronization-Buffering, Process Addressing, Failure Handling, Remote procedure call (RPC): Model, Implementation, Stub generation, RPC messages, Marshaling, server Management, Call semantic, communication protocols for RPC-Client server binding, RMI.

PRACTICES:

- Simulate the Distributed Mutual exclusion.
- Simulate the Non Token/ Token based algorithm in Distributed system.
- Implement 'RPC' mechanism for accessing methods of remote systems

MODULE-2

UNIT-1 **8L+8T+0P=16 Hours**

DISTRIBUTED SHARED MEMORY:

Distributed shared memory- Introduction; Synchronization: Synchronizing physical clocks, Logical clocks, Distributed File Systems: Introduction, File Models, File accessing, sharing and caching, File Replication, Atomic transactions.

UNIT-2 **8L+8T+0P=16Hours**

DISTRIBUTED PROCESSES:

Distributed Shared Memory, Design and implementation issues, Sequential consistency, Release consistency, Process migration Features & Mechanism; Distributed coordination, Event Ordering, Mutual Exclusion, Deadlock, Election algorithms. Resource and process management, Task assignment approach, Load balancing approach, Load sharing approach, Case Study HADOOP

SKILLS:

- ✓ *Design and simulate Distributed Mutual Exclusion.*
- ✓ *Implement the RPC in chat applications.*
- ✓ *Apply the Distributed File system concepts*

PRACTICES:

- Experiment with Message Passing Interface Standard (MPI).
- Case study on Hadoop.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the knowledge of distributed systems fundamentals, goals, and evolution to design simple distributed applications.	PO1, PO2, PO3, PSO1
CO2	Apply communication protocols, message passing, RPC, and RMI for solving real-time problems in distributed environments.	PO2, PO3, PSO1, PSO3
CO3	Analyze distributed shared memory, synchronization, file systems, and migration techniques.	PO2, PO4, PSO2
CO4	Evaluate coordination mechanisms like mutual exclusion, deadlock handling, election, and distributed resource management.	PO2, PO4, PO5, PO12, PSO2, PSO3

TEXT BOOK:

1. George Colouris, Jean Dollimore and Tim Kindberg, “Distributed Systems – Concepts and Design”, Pearson Education Private Limited, 2001.

REFERENCE BOOKS:

1. Tanenbaum, A.S. and Van Steen M., “Distributed systems: principles and paradigms”. Prentice-Hall, 2007.
2. Sinha, P.K., “Distributed operating systems: concepts and design”, PHI Learning Pvt. Ltd, 1998
3. Liu, M.L., “Distributed computing: principles and applications”, Pearson Education Inc, , 2003.

22IT821 MICROPROCESSORS AND MICROCONTROLLERS

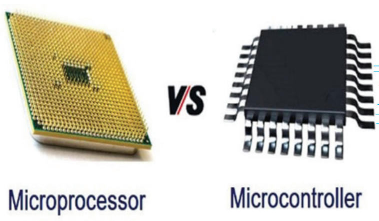


Image Source: https://components101.com/sites/default/files/components/Difference-between-Microprocessor-and-Microcontroller.jpg

Hours Per Week :

L	T	P	C
3	2	-	4

PREREQUISITE KNOWLEDGE: Basics of digital design and logic

COURSE DESCRIPTION AND OBJECTIVES:

This course deals with the basic architecture, assembly language programming, pin definitions, supporting chips and memory interfacing of microprocessors and microcontrollers. The objective of the course is to understand various addressing modes, different peripheral devices and their interfacing with 8086, 8051 and ARM processor.

MODULE-1

UNIT-1 **12L+8T+0P=20 Hours**

INTRODUCTION TO 8086 MICROPROCESSOR:

Evolution of microprocessors; 8086 microprocessor Architecture, Register model, Memory segmentation, Physical address generation, Addressing modes, Instruction set, Interrupts of 8086. Pin configuration of 8086; 8086 system bus architecture, Physical Memory organization Systems.

UNIT-2 **12L+8T+0P=20 Hours**

INTRODUCTION TO 8051 MICROCONTROLLER:

Comparing Microprocessors and microcontrollers; 8051 Micro controller Architecture; Signal Description of 8051; Memory organization; Addressing modes of 8051; Instruction set; Assembly language program examples in 8051.

PRACTICES:

- Programs on different Data Transfer Instructions using 8086.
- Arithmetic operations: Addition, Subtraction, Multiplication, Division using 8086.
- Programs to analyze different addressing modes of 8086.
- Program to sort the array of given numbers in ascending order.

MODULE-2

UNIT-1 **12L+8T+0P=20 Hours**

8051 MICROCONTROLLER HARDWARE AND PERIPHERAL INTERFACING:

Parallel Ports in 8051; 8051 Timers; 8051 Serial ports; 8051 Interrupts. Peripheral Interfacing- LCD and Keyboard Interfacing, ADC and Sensor Interfacing, DC Motor and Stepper Motor Interfacing Techniques.

UNIT-2 **12L+8T+0P=20 Hours**

ARM ARCHITECTURE:

RISC Vs CISC systems – ARM Philosophy – ARM7TMI Core Architecture, Functional Diagram, Programmer’s Model, ARM State Register Set, HUMB state register set, Current Program Status Register, ARM 7TMI Operating modes, mode bits, Exceptions, Interrupt Vector Table, Interrupt Processing.

SKILLS:

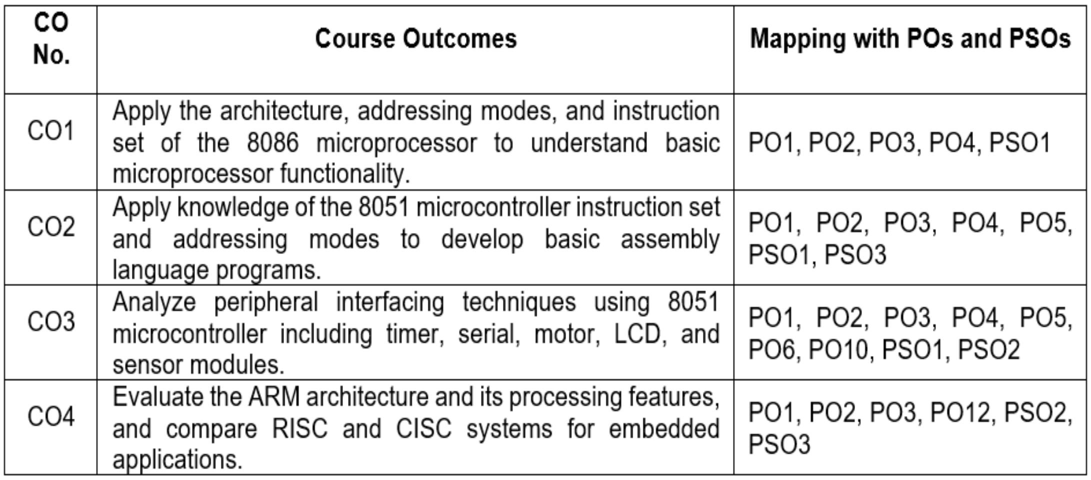
- ✓ *Develop assembly language programs for 8086 Microprocessor and 8051 Microcontroller.*
- ✓ *Design microprocessor based system for given applications.*

PRACTICES:

- Interfacing 7 Segment LED Display to 8051
- Alphanumeric LCD panel interface to 8051.
- Hex keypad input interface to 8051.
- ADC interface to 8051.
- DAC interface to 8051 for waveform generation.
- Stepper motor control interface to 8051.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the architecture, addressing modes, and instruction set of the 8086 microprocessor to understand basic microprocessor functionality.	PO1, PO2, PO3, PO4, PSO1
CO2	Apply knowledge of the 8051 microcontroller instruction set and addressing modes to develop basic assembly language programs.	PO1, PO2, PO3, PO4, PO5, PSO1, PSO3
CO3	Analyze peripheral interfacing techniques using 8051 microcontroller including timer, serial, motor, LCD, and sensor modules.	PO1, PO2, PO3, PO4, PO5, PO6, PO10, PSO1, PSO2
CO4	Evaluate the ARM architecture and its processing features, and compare RISC and CISC systems for embedded applications.	PO1, PO2, PO3, PO12, PSO2, PSO3

TEXT BOOKS:

1. Douglas V.Hall, “Microprocessors and Interfacing”, 3rd edition, Tata McGraw Hill, 2017.
2. Kenneth J. Ayala, “The 8051 Microcontroller”, 3rd edition, Cengage Learning India Pvt. Ltd, 2010.

REFERENCE BOOKS:

1. Barry B. Brey, “The Intel microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro processor, Pentium II, Pentium III, Pentium 4, and Core2 with 64-bit extensions:architecture, programming, and interfacing”, 8th edition, Pearson Prentice Hall, 2009.
2. Mohamed Rafiquzzaman, “Microprocessors and Microcomputer Based System Design”, 2nd edition, CRC Press, 2007.
3. Steve Furber, “ARM System on Chip Architecture”, 2nd edition, Pearson education, 2000.

INFORMATION TECHNOLOGY

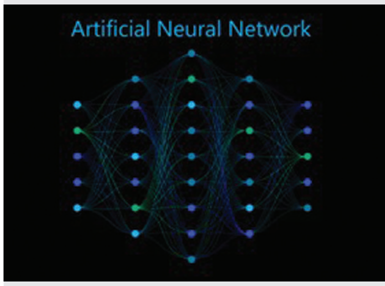
B.Tech.

Honors

COURSE CONTENTS

I SEM & II SEM

22IT951 ARTIFICIAL NEURAL NETWORKS



Source: https://media.istockphoto.com/vectors/neural-network-model-with-thin-synapses-between-neurons-vector-id1156059931?k=20&m=1156059931&s=612x612&w=0&h=UEkHaO0BFXlsHv6PDHxzKgo2fKJexkqtBDt-ZRAyw3ml=

Hours Per Week :

L	T	P	C
3	2	-	4

PREREQUISITE KNOWLEDGE: Basics of algorithms and programming

COURSE DESCRIPTION AND OBJECTIVES:

The course introduces the principles of neuro-computing with artificial neural networks, which widely used for addressing real-world problems such as classification, regression, pattern recognition, data mining, time-series modelling, etc.. This course majorly covers two kinds of learning such as supervised and unsupervised.

MODULE-1

UNIT-1 **12L+8T+0P=20 Hours**

ARCHITECTURE OF NEURAL NETWORK:

Introduction to Artifical Neural Networks: Introduction; Artificial neural networks; Historical development of neural networks; Biological Neural Networks; Basic building blocks; Terminologies. Fundamental Models of Artificial Neural Networks: Introduction, McCullochPitts Neuron model; Perceptron Networks: Introduction; Single layer perceptron; Brief introduction to multilayer perceptron networks.

UNIT-2 **12L+8T+0P=20 Hours**

MODELS OF ARTIFICIAL NEURAL NETWORKS:

Comparison between them and the Computer; Comparison between artificial and biological neural network; Fundamental Models of Artificial Neural Networks: Learning rules, Hebbian learning rule, perceptron learning rule, Delta learning rule, Widrow-H off rule or least mean square (LMS) rule, Competitive learning rule; Out star learning; Boltzmann based learning; Hebb net.

PRACTICES:

- Plot a few activation functions that are being used in neural networks.
- Generate ANDNOT function using McCulloch-Pitts neural net.
- Generate XOR function using McCulloch-Pitts neuron.
- With a suitable example demonstrate the perceptron learning law with its decision regions.

MODULE-2

UNIT-1 **12L+8T+0P=20 Hours**

TYPES OF NEURAL NETWORKS:

Adaline and Madaline Networks: Introduction, Adaline, Madaline. Associative Memory Networks: Introduction; Feedback Networks: Introduction; Feed Forward Networks: Introduction; Self Organizing Feature Map: Introduction; Adaptive Resonance Theory: Introduction, ART Fundamentals, ART1, ART2.

UNIT-2 **12L+8T+0P=20 Hours**

ADVANCED NEURAL NETWORKS:

Adaline and Madaline Networks: Algorithms for pattern association, Hetero associative memory neural networks, Auto associative memory network, Bi-directional associative memory. Feedback Networks: Discrete Hopfiled Net, Continuous Hopfiled Net, Relation between BAM and Hopfiled Nets. FEED Forward Networks: Back Propagation Network (BPN), Radial Basis Function Network (RBFN).

SKILLS:

- ✓ *Create different neural networks of various architectures both feed forward and feed backward.*
- ✓ *Perform the training of neural networks using various learning rules.*
- ✓ *Testing of neural networks and do the perform analysis of these networks for various pattern recognition applications.*

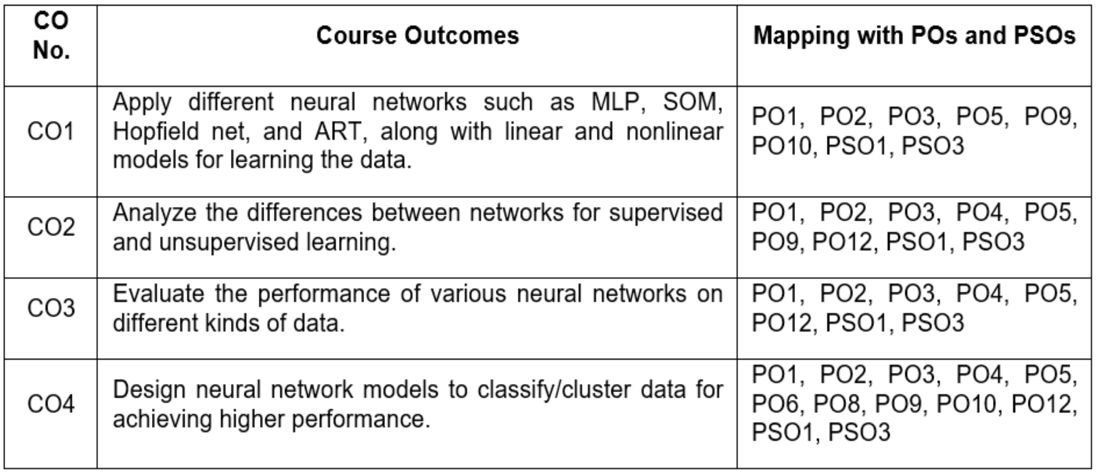
SelfOrganizing Feature Map: Introduction; Methods used for determining the winner; Kohonen self-organizing feature maps; Learning vector quantization(LVQ); MaxNet, Maxican Hat, Hamming Net.

PRACTICES:

- Program to show Back Propagation Network for XOR function with Binary Input and Output.
- Program to show Back Propagation Network for XOR function with Bipolar Input and Output.
- Program to illustrate ART neural network.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply different neural networks such as MLP, SOM, Hopfield net, and ART, along with linear and nonlinear models for learning the data.	PO1, PO2, PO3, PO5, PO9, PO10, PSO1, PSO3
CO2	Analyze the differences between networks for supervised and unsupervised learning.	PO1, PO2, PO3, PO4, PO5, PO9, PO12, PSO1, PSO3
CO3	Evaluate the performance of various neural networks on different kinds of data.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1, PSO3
CO4	Design neural network models to classify/cluster data for achieving higher performance.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3

TEXT BOOK:

1. Sivanandam, S.Sumathi and S.N.Deepa; “Introduction to Neural Networks”, 2nd Edition., TATA McGraw HILL: 2005.

REFERENCE BOOKS:

1. Simon. Hhaykin, “Neural networks A comprehensive foundation”, 2nd Edition, Pearson Education, 2004.
2. B. Yegnanarayana, “Artificial neural networks”, 1st Edition., Prentice Hall of India Pvt. Ltd, 2005.
3. Li Min Fu, “Neural networks in Computer intelligence”,1st Edition., TMH, 2003.

22IT953 DEEP LEARNING

Hours Per Week :

L	T	P	C
3	2	-	4

PREREQUISITE KNOWLEDGE: Basics of data mining, Artificial neural networks, Python & R programming

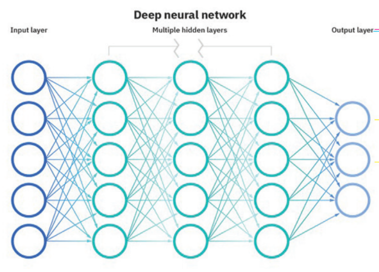


Image source: https://1.cms.s81c.com/sites/default/files/2021-01-06/ICLH_Diagram_Batch_01_03-DeepNeuralNetwork-WHITEBG.png

COURSE DESCRIPTION AND OBJECTIVES:

The objective of this course is to introduce major deep learning algorithms, the problem settings, and their applications to solve real world problems. The course also requires students to implement programming assignments related to these topics such as recurrent neural networks, long short-term memory cells and convolution neural networks.

MODULE-1

UNIT-1 **12L+8T+0P=20 Hours**

FUNDAMENTAL CONCEPTS OF DEEP LEARNING:

Introduction to Deep Learning: Deep Learning, Paradigms of learning problems, Types of Deep Neural Networks – Artificial Neural Network, Multi-layer neural network, Convolution neural network, Recurrent neural network, Deep belief network, Perspectives, and Issues in deep learning framework.

Applications of Deep Learning – Fraud detection, Customer relationship management systems, Computer vision, Natural language processing, Autonomous vehicles, E-commerce.

UNIT-2 **12L+8T+0P=20 Hours**

TYPES OF DEEP NEURAL NETWORKS:

Convolution Neural Networks – Basic structure of Convolutional Network, Case studies: Alex net, VGGNet, GoogLeNet, Applications of CNN– Object Detection, Content based image Retrieval.

Recurrent Neural Networks – Basic structure of RNN, Case studies: Long Short-Term Memory, Gated Recurrent Units, Back propagation through time, Bidirectional LSTMs, and Bidirectional RNNs, Applications of RNN– Language Modelling and Generating Text, Text Summarization.

PRACTICES:

- Develop a deep learning model with an Artificial Neural Network to create a predictive model based on the Churn dataset for new customers. The predictive model must predict for any new customer that he or she will stay in the bank or leave the bank. So that bank can offer something special for the customers whom the predictive model predicts will leave the bank.
- Develop a deep learning model of your First Image Processing Project with Convolutional Neural Network.
- Develop a deep learning model for fruit classification using GoogleNet Convolutional Neural Network.
- Develop a deep learning model to solve sequence prediction problem using RNN with Dropout condition.
- Develop a deep learning model to perform anomaly detection model using LSTM.

MODULE-2

UNIT-1 **12L+8T+0P=20 Hours**

GENERATIVE & ATTENTION MODELS:

Generative models – Restrictive Boltzmann Machines (RBMs), Deep Boltzmann Machines, Autoencoders, and Generative Adversarial Networks.

Attention and memory models – Attention Mechanism, Types of Attention models – Local Attention, Global Attention, and Self-Attention.

UNIT-2 **12L+8T+0P=20 Hours**

BETTER TRAINING OF DEEP NEURAL NETWORKS:

Training Neural Network – Keras, Tensorflow, Regularizations, Training deep models, Dropouts, Multi-task learning, Early Stopping, Parameter Initialization and Parameter Sharing, Newer optimization methods for neural networks (Adagrad, adadelta, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization).

PRACTICES:

- Develop a project for implementing Image Segmentation using CNN with Tensorflow.
- Develop a project for implementing Time Series Forecasting with LSTM Neural Network Python.
- Develop a project for building a Review Classification Model using Gated Recurrent Unit.
- Develop a project for building CNN Image Classification Models for Real Time Prediction.
- Design a Deep Learning Project on Store Item Demand Forecasting.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply forward and backward propagation concepts in different types of Artificial Networks such as MLP, CNN, RNN and Apply Pre-Trained models to perform classification tasks.	PO1, PO2, PO3, PO5, PO9, PO10, PO12, PSO1, PSO3
CO2	Evaluate the differences among MLP, CNN, and RNN functioning and prioritize the models along with different parameters (learning rate, activation function, batch size, epochs, optimizers, etc.) for different problems.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO12, PSO1, PSO3
CO3	Analyze the relationship and limitations of classification, different model evaluation metrics.	PO1, PO2, PO3, PO5, PO9, PO10, PO12, PSO1, PSO3
CO4	Develop image and text classifier models for various real-time applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3

TEXT BOOKS:

1. CharuC.Aggarwal, "Neural Networks and Deep learning", Springer International Publishing, 2018
2. Satish Kumar, "Neural Networks, A Classroom Approach", Tata McGraw -Hill, 2007.

REFERENCE BOOKS:

1. P.A. Devijer & J. Kittler, "Pattern Recognition - A Statistical Approach", Prentice-Hall, 2008.
2. Christopher. M. Bishop, "Pattern recognition and machine learning", Springer, 2006.
3. M. NarasimhaMurthy and Der V Susheela Devi, "Introduction to Pattern Recognition and Machine Learning", IISC Press, 2008.

SKILLS:

- ✓ *To analyze the fundamentals of deep learning.*
- ✓ *Main techniques in deep learning and the main research in this field.*
- ✓ *Design and implement deep neural network systems.*
- ✓ *Identify new application requirements in the field of computer vision.*

22IT952 COMPUTER VISION

Image source: https://cdn.nextgov.com/media/img/cd/2020/11/03/NGai20201103/860x394.jpg

Hours Per Week :

L	T	P	C
3	2	-	4

PREREQUISITE KNOWLEDGE: Artificial intelligence and Machine learning.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the development of algorithms and techniques to analyze and interpret the visible world around us. This requires an understanding of the fundamental concepts related to multi-dimensional signal processing, feature extraction, pattern analysis visual geometric modeling, stochastic optimization, etc. Knowledge of these concepts is necessary for this field, to explore and contribute to research and further developments in the field of computer vision. Applications range from Biometrics, Medical diagnosis, document processing, and mining of visual content, to surveillance, advanced rendering, etc.

MODULE-1

UNIT-1 **12L+6T+0P=18 Hours**

COMPUTER VISION AND TRANSFORMATION:

Computer Vision: Introduction to computer vision, Digital Image Formation and low-level processing, Overview and State-of-the-art, Fundamentals of Image Formation.

Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution, Filtering, Image Enhancement, Restoration, Histogram Processing.

Image Formation: Geometric primitives and transformations, 2D transformations, 3D transformations, 3D rotations, 3D to 2D projections.

UNIT-2 **12L+10T+0P=22 Hours**

PHOTOMETRIC IMAGE FORMATION, DIGITAL CAMERA AND IMAGE PROCESSING:

Photometric Image Formation: Lighting, Reflectance and shading, Optics.

The Digital Camera: Sampling and aliasing, Color, Compression.

Image Processing: Point operations- Pixel transforms, Color transforms, Compositing and matting, Histogram equalization. Linear filtering – Separable filtering, Band-pass and steerable filters. More neighbourhood operators – nonlinear filtering, Bilateral filtering, Binary image processing. Fourier transforms, Pyramids and wavelets, and Geometric transformations. Model fitting and optimization.

PRACTICES:

- Digital image formation and low level processing.
- Orthogonal, Euclidean, Affine and Projective.
- Filtering, Image Enhancement, Restoration, Histogram Processing.
- 2D transformations, 3D transformations.
- 3D rotations, 3D to 2D projections.
- Reflectance and shading.
- Color and compression.
- Pixel transforms, Color transforms.
- Histogram equalization, Linear filtering.
- Fourier transforms, Pyramids and wavelets.

SKILLS:

- ✓ *Develop image analysis algorithms.*
- ✓ *Develop Deep Learning frameworks to solve problems of computer vision.*
- ✓ *Design and create platforms for image processing and visualization.*

MODULE-2

UNIT-1 **12L+6T+0P=18 Hours**

RECOGNITION, FEATURE DETECTION AND MATCHING, IMAGE ALIGNMENT:

Recognition: Instance recognition, Image classification, Object detection, Semantic segmentation, Video understanding, Vision and language.

Feature Detection and Matching: Points and Patches, Edges and contours, Contour tracking, Line and vanishing points, and segmentations.

Image Alignment and Stitching: Pairwise alignment, Image stitching, Global alignment, and Compositing.

Motion Estimation: Translational alignment, Parametric motion, Optical flow, Layered motion.

UNIT-2 **12L+10T+0P=22 Hours**

COMPUTATIONAL PHOTOGRAPHY:

Computational Photography: Photometric calibration, High dynamic range imaging, Super resolution denoising and blur removal, Image matting and compositing, Texture analysis and synthesis.

Depth Estimation: Epipolar geometry, Sparce correspondence, Dense correspondence, Local methods, Global optimization, Depth neural network.

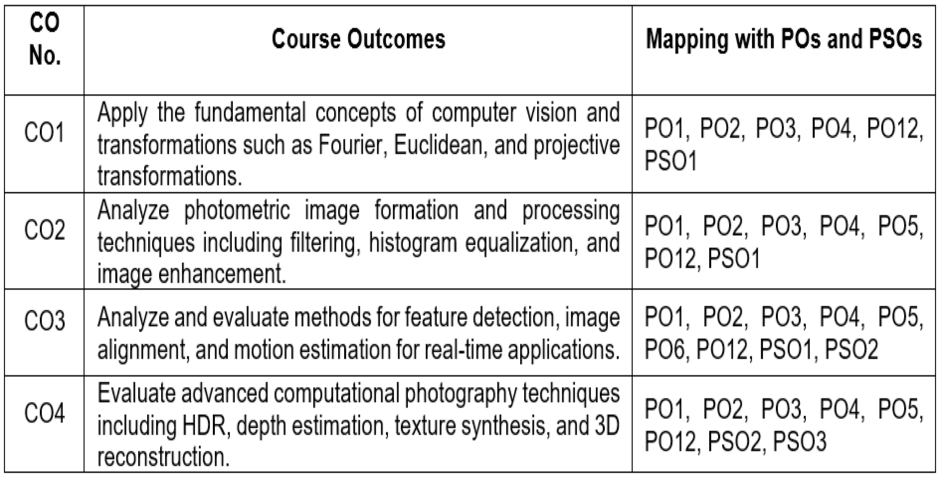
3D Reconstruction: Shape form X, 3D scanning, Surface representation, Point based representation.

PRACTICES:

- Image classification, Object detection.
- Points and Patches, Edges and contours.
- Global alignment and compositing.
- Parametric motion and optical flow.
- Photometric calibration, High dynamic image range.
- Super resolution denoising.
- Epipolar geometry.
- Global optimization.
- 3D scanning, Surface representation.
- Point based representation.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply the fundamental concepts of computer vision and transformations such as Fourier, Euclidean, and projective transformations.	PO1, PO2, PO3, PO4, PO12, PSO1
CO2	Analyze photometric image formation and processing techniques including filtering, histogram equalization, and image enhancement.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1
CO3	Analyze and evaluate methods for feature detection, image alignment, and motion estimation for real-time applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO2
CO4	Evaluate advanced computational photography techniques including HDR, depth estimation, texture synthesis, and 3D reconstruction.	PO1, PO2, PO3, PO4, PO5, PO12, PSO2, PSO3

TEXT BOOK:

1. Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer-Verlag London Limited, 2021.

REFERENCE BOOKS:

1. Richard Hartley and Andrew Zisserman, “Multiple View Geometry in Computer Vision”, 2nd Edition, Cambridge University Press, March 2004.
2. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006
3. R.C. Gonzalez and R.E. Woods, “Digital Image Processing, Addison”, Wesley, 1992.
4. K. Fukunaga; Introduction to Statistical Pattern Recognition, 2nd Edition, Academic Press, Morgan Kaufmann, 1990.

Image source: https://www.einfochips.com/blog/role-video-analytics-smart-city-traffic-control/

22IT954 IMAGE AND VIDEO ANALYTICS

Hours Per Week :

L	T	P	C
3	2	-	4

PRPREREQUISITE KNOWLEDGE: Basics of machine learning and Image processing

COURSE DESCRIPTION AND OBJECTIVES:

This course explores and demonstrate real time image and video analytics in solving practical problems of commercial and scientific interests. The objective of this course is to offer the knowledge and skill on the basic principles and concepts in digital image and video processing.

MODULE-1

UNIT-1 **12L+8T+0P=20 Hours**

IMAGE PROCESSING:

Python Installation and Working of it, get familiar with python variables and data types, Operator understanding and its usage, detail study of python blocks, Hands on with conditional blocks using if, else and elif

UNIT-2 **12L+8+0P=20 Hours**

FEATURE EXTRACTION AND TEXTURE ANALYSIS:

Feature Extraction: Binary object feature, Histogram-based (Statistical) Features, Intensity features, Shape feature extraction, PCA - SIFT – SURF. Texture Analysis - Concepts and classification, statistical, structural and spectral analysis.

Object Recognition: Patterns and pattern class, Bayes' Parametric classification, Feature Selection and Boosting, Template- Matching. Content Based Image Retrieval - Feature based image retrieval, Object Based Retrieval.

PRACTICES:

- Video surveillance

  Description:

  Crowd management can provide an estimated crowd count in real time and trigger alerts when a threshold is reached or surpassed. They can also analyze crowd flow to detect movement in unwanted or prohibited directions.
- Traffic analysis

  Description:

  Vehicle counting, or differentiating between cars, trucks, buses, taxis, and so on, generates high-value statistics used to obtain insights about traffic. Installing speed cameras allows for precise control of drivers. Automatic license plate recognition identifies cars that commit an infraction or, thanks to real-time searching, spots a vehicle that has been stolen or used in a crime.

MODULE-2

UNIT-1 **12L+8+0P=20 Hours**

VIDEO TRACKING:

Digital Video, Sampling of video signal, Video Enhancement and Noise Reduction- Rate control and buffering, MPEG, H.264, Inter frame Filtering Techniques, Fundamentals of Motion Estimation and Motion Compensation.

Change Detection, Background modelling, Motion Segmentation, Simultaneous Motion Estimation and Segmentation, Motion Tracking, Multi-target/Multi-camera tracking.

UNIT-2 **12L+8+0P=20 Hours**

VIDEO ANALYSIS ACTION RECOGNITION:

Video Analysis Action Recognition, Video based rendering, Context and scene understanding. Case Study: Surveillance - Advanced Driver Assistance System.

PRACTICES:

- Smart Encoder Appliance.
- Self Service Portal.
- Smart Search Example: "person wearing blue"
- Video Summarization.
- Suspect Identification.
- Real-Time Alerts.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply Python programming to perform basic image processing tasks and control structures.	PO1, PO2, PO3, PO12, PSO1, PSO3
CO2	Analyze features and textures in images using various extraction and classification techniques.	PO1, PO2, PO4, PO5, PSO1, PSO3
CO3	Apply motion estimation and tracking techniques for video processing in real-time scenarios.	PO1, PO2, PO3, PO4, PO5, PSO1, PSO3
CO4	Evaluate action recognition techniques and perform video analysis for advanced applications like surveillance and driver assistance.	PO1, PO2, PO5, PO10, PSO1, PSO3

TEXT BOOKS:

1. Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", 3rd Edition, Prentice-Hall, 2008.
2. Murat Tekalp, "Digital Video Processing", 2nd Edition, Prentice Hall, 2015.

REFERENCE BOOKS :

1. Oge Marques," Practical Image and Video Processing Using MATLAB", Wiley-IEEE Press, 2011.
2. Yu Jin Zhang, "Image Engineering: Processing, Analysis and Understanding", Tsinghua University Press, 2009.
3. Mark Nixon and Alberto S. Aquado, "Feature Extraction & Image Processing for Computer Vision", 3rd Edition, Academic Press, 2012.

SKILLS:

- ✓ *Able to apply various fundamentals for image processing.* .
- ✓ *Able to implement video analysis action recognition.*
- ✓ *Able to identify and recognize object in motion.*

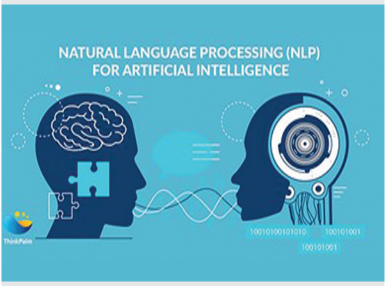

NATURAL LANGUAGE PROCESSING (NLP)
FOR ARTIFICIAL INTELLIGENCE
ThinkPalm

Image source: https://thinkpalm.com/wp-content/uploads/2019/04/BLOG_NLP-FOR-ARTIFICIAL-INTELLIGENCE_72-1.jpg

22IT955 NATURAL LANGUAGE PROCESSING

Hours Per Week :

L	T	P	C
3	2	-	4

PREREQUISITE KNOWLEDGE: Basics of classification techniques and statistical measures.

COURSE DESCRIPTION AND OBJECTIVES:

The goal of this course is to provide knowledge on fundamental concepts of Natural Language Processing. The methods for representing and classifying the written text is the primary focus. Students will learn tokenization, word representation (TF-IDF and word vectors) and classification techniques. Further understanding sentiment analysis and recommendation systems.

MODULE-1

UNIT-1 **12L+8+0P=20 Hours**

N GRAM LANGUAGE MODELS:

Words, Corpora, Text Normalization, Minimum Edit distance. N Grams, Evaluating Language Models, Generalization and Zeros, Smoothing. Naïve Bayes and Sentiment Classification,

UNIT-2 **12L+8+0P=20 Hours**

SENTIMENT ANALYSIS:

Training the Naïve Bayes classifier, Optimizing for sentiment analysis, Naïve Bayes for other text classification tasks, as a language model, Evaluation Test sets and Cross validation, Statistical significance of testing, advanced feature selection.

PRACTICES:

- Computing 1gram,2gram and N gram tokens.
- Stop word removing.
- Stemming and Lemmatization.
- Performing sentiment analysis using Naïve Bayes.
- Building a language model.
- Latent Semantic Analysis for feature selection/dimensionality reduction.
- Latent Dirichlet Allocation method for topic modelling.

MODULE-2

UNIT-1 **12L+8+0P=20 Hours**

VECTOR SEMANTICS AND SEQUENCE PROCESSING:

Vector Semantics: Lexical semantics, vector semantics, word vectors, Cosine for measuring similarity, TF-IDF: weighting terms in the vector, Applications, Word2Vec, Visualizing Embeddings, semantic properties of embeddings, Evaluating Vector models. Sequence Processing with Recurrent Networks: Simple Recurrent Networks, Applications of RNN's, Deep Networks: Stacked and Bidirectional RNNs, managing Context in RNNs: LSTMs and GRU.

UNIT-2 **12L+8+0P=20 Hours**

POS TAGGING:

Part of speech tagging: English Word classes, The Penn Treebank POS tag set, POS tagging, HMM POS tagging, Maximum Entropy Markov Models.

PRACTICES:

- Computing word similarity using cosine distance.
- Computing TF-IDF values for words in a corpus.
- Computing word embeddings using word2vec.
- Visualizing the word vectors using TSNE plots.
- Tokenizing and sequence generation.
- RNN based sentiment analysis.
- LSTM based sentiment analysis.
- GRU based sentiment analysis.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply foundational NLP techniques including tokenization, normalization, stemming, lemmatization, and construct language models using N-grams, Naïve Bayes classifiers, TF-IDF, and topic modeling.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO12, PSO1, PSO3
CO2	Analyze text data using sentiment analysis, part-of-speech tagging, and word embedding models like Word2Vec, and evaluate them with metrics such as accuracy, confusion matrix, ROC curves, and statistical validation techniques..	PO1, PO2, PO3, PO4, PO5, PO9, PO12, PSO1, PSO3
CO3	Evaluate the effectiveness of NLP models N-gram, Naïve Bayes, vector space, topic modeling, and deep learning (RNN, LSTM, GRU)—by assessing their performance in classification, feature extraction, and sequence modeling tasks.	PO1, PO2, PO3, PO4, PO5, PO12, PSO1, PSO3
CO4	Create complete NLP solutions by designing and implementing language models, sentiment classifiers, vector embeddings, sequence tagging models (using RNNs, LSTMs, GRUs), and topic models (LSA, LDA) for real-world textual data	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO12, PSO1, PSO3

TEXT BOOK:

1. Daniel Jurafsky, James H. Martin, “An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition”, 3rd Edition, Pearson Publication, 2014.

REFERENCE BOOK:

1. Steven Bird, Ewan Klein and Edward Loper, “Natural Language Processing with Python”, 1st Edition, O_Reilly Media, 2009.

SKILLS:

- ✓ *Text pre-processing and Analyzing.*
- ✓ *Develop sentiment analyzers.*
- ✓ *Develop recommendation systems.*
- ✓ *Computing similarity score.*
- ✓ *Develop Context preserving language models.*

22IT956 CODING COMPETENCY

Hours Per Week :

L	T	P	C
-	4	4	4

PREREQUISITE KNOWLEDGE: Basic Programming Knowledge

COURSE DESCRIPTION AND OBJECTIVES:

This course is designed to equip students with the foundational and advanced skills needed for programming, data structures, algorithm design, and database integration. The course provides extensive hands-on practice in solving real-world problems using efficient data structures and algorithms, and it includes practical sessions on writing SQL queries and connecting databases to applications. The course aims to prepare students for software development roles and technical interviews by strengthening their coding and database handling capabilities.

MODULE – 1

UNIT-1:

Basic Programming & Data Structures

Programming Basics: Variables, Operators, Conditionals, Loops, Functions, Pointers and Arrays in C, Strings and Matrix Manipulations, Linked Lists (Singly, Doubly, Circular), Stack and Queue – Applications in Expressions, Recursion, Scheduling

Hands-on Practice:

UNIT- 2:

Advanced Data Structures for Competitive Coding

Trees: Binary Trees, Binary Search Trees, AVL Trees, Graphs: Representation, BFS/DFS, Pathfinding, Hashing: Collision resolution techniques, HashMaps, and Sets

Practices:

- Writing functions and using pointers
- Debugging linked list operations
- Expression evaluation using stacks
- Coding challenges on arrays and strings
- Tree traversals and applications
- Graph problems from online judges (e.g., LeetCode, Codeforces)
- Hashing-based search and pattern-matching problems

MODULE-2

UNIT-1:

Problem Solving with Data Structures

Sliding Window, Two Pointer, and Greedy Techniques, Recursion and Backtracking, Sorting and Searching Algorithms for Large Datasets

UNIT 2:

Design and Analysis of Algorithms

Algorithm Design Paradigms: Divide & Conquer, Greedy, Dynamic Programming, Time and Space Complexity Analysis, Case Study: Solving real-world problems using multiple paradigms

Practices:

- Implementing and comparing sorting algorithms
- Real-world case studies using search and optimization
- Daily coding problem-solving sessions
- Solve classical problems (e.g., Knapsack, Matrix Chain Multiplication)
- Compare algorithm efficiency
- Mock coding tests with complexity analysis

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Implement and apply fundamental programming and data structures in solving real-time problems	PO1, PO2, PO5, PSO1, PSO3
CO2	Solve competitive coding problems using optimal data structure strategies	PO1, PO2, PO3, PSO1, PSO3
CO3	Design efficient algorithms and analyze time-space complexity Evaluate/Create	PO4, PO5, PO6, PSO1, PSO3
CO4	Construct and integrate SQL databases into real-world applications	PO3, PO4, PO6, PO12, PSO1, PSO3

TEXT BOOK:

1. Narasimha Karumanchi, Data Structures and Algorithms Made Easy, CareerMonk Publications
2. Thomas H. Cormen, Introduction to Algorithms, MIT Press
3. Silberschatz, Korth, Sudarshan, Database System Concepts, McGraw Hill

REFERENCE BOOK:

1. GeeksforGeeks, LeetCode, HackerRank (Coding Platforms)
2. Online docs (MySQL/PostgreSQL), Real Python (Database APIs)

22IT957 ADVANCED WEB TECHNOLOGIES

Hours Per Week :

L	T	P	C
-	4	4	4

PREREQUISITE KNOWLEDGE: Web Technologies

COURSE DESCRIPTION AND OBJECTIVES: This course introduces students to modern web development technologies and DevOps practices essential for full-stack application development and deployment. Students will learn client-side development using ReactJS, server-side programming with Node.js, and continuous integration and deployment strategies using DevOps tools. Through hands-on sessions and project-based learning, students will gain practical experience in building, deploying, and managing dynamic web applications.

MODULE-1

UNIT 1

Basic Web Technologies and ReactJS Basics

Overview of Web Development Architecture, Introduction to ReactJS and JSX, Components, Props, and State, React Lifecycle Methods and Hooks

UNIT **2:**

Advanced ReactJS Development

Routing using React Router, Forms, Controlled Components, and Validation, State Management using Redux, REST API Integration

PRACTICES:

- Building static and dynamic components
- Creating responsive UIs with stateful components
- Developing SPA (Single Page Applications)
- Connecting React app with APIs using Axios or Fetch

MODULE-2

UNIT 1:

Node.js and Backend Development

Introduction to Node.js and NPM, Building servers using Express.js, Working with MongoDB and Mongoose, User Authentication and Middleware

UNIT 2:

DevOps Fundamentals

Introduction to DevOps and SDLC, Version Control using Git and GitHub, CI/CD Pipelines using Jenkins or GitHub Actions, Containerization with Docker

Practices:

- Creating backend APIs
- Performing CRUD operations and validations
- Automating builds and deployments
- Dockerizing web applications

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Design interactive user interfaces using ReactJS components and state handling	PO1, PO2, PO5, PSO1, PSO3
CO2	Develop and test RESTful APIs using Node.js and Express	PO1, PO2, PO3, PSO1, PSO3
CO3	Apply DevOps tools and practices to automate software delivery pipelines	PO4, PO5, PO6, PSO1, PSO3
CO4	Deploy, monitor, and manage full-stack applications on cloud platforms	PO3, PO4, PO6, PO12, PSO1, PSO3

TEXTBOOK:

- Robin Wieruch, *The Road to React*, 2023 Edition
- Ethan Brown, *Web Development with Node and Express*, O'Reilly Media
- Ernest Mueller & James Wickett, *DevOps Fundamentals*, O'Reilly Media

REFERENCE BOOK:

- ReactJS Official Documentation: https://reactjs.org
- Node.js Documentation: https://nodejs.org
- Docker & GitHub Docs: https://docs.docker.com, https://docs.github.com

22IT963 DEVOPS TECHNOLOGIES

Hours Per Week :

L	T	P	C
-	4	4	4

PREREQUISITE KNOWLEDGE: Web Technologies, Cloud Computing

COURSE DESCRIPTION AND OBJECTIVE: This course introduces students to the practical tools and cloud-native technologies used in DevOps workflows. Students will gain hands-on experience with AWS services like EC2, S3, IAM, and CloudWatch, containerization using Docker, and automation of application deployment using CI/CD pipelines. The course bridges development and operations using real-world examples, enhancing students' ability to build scalable, maintainable, and secure cloud applications.

MODULE – 1

UNIT–1:

AWS Infrastructure Setup
Launching EC2 Instances, Creating and Managing S3 Buckets, Configuring IAM Roles and Policies, Introduction to VPC and Security Groups

UNIT–2:

Docker and Containers
Docker Installation, Creating Docker Images and Containers, Docker Hub Integration, Container Networking, Docker Compose

PRACTICES:

- Launch a Linux EC2 instance and install a web server
- Create and configure an S3 bucket, set access permissions, and enable static website hosting
- Create IAM users and attach custom roles and policies
- Use AWS CLI to manage S3 buckets and EC2 instances
- Build and run a Docker container for a sample Node.js or Python application
- Push/pull Docker images to/from Docker Hub
- Create and manage multi-container apps using Docker Compose

MODULE-2

UNIT–1:

CI/CD Pipelines
Introduction to DevOps Pipelines, CodeCommit (Git), CodeBuild, CodeDeploy,CodePipeline, GitHub Actions Integration

UNIT–2:

Monitoring and Automation
CloudWatch Metrics and Alarms, CloudTrail Logs, Auto Scaling Groups, Load Balancer Configuration, Infrastructure as Code (IaC) with AWS CloudFormation

PRACTICES:

- Create a CI/CD pipeline using AWS CodePipeline to deploy a web app
- Integrate GitHub repository with AWS CodeBuild and CodeDeploy
- Configure CloudWatch to monitor EC2 instance health and set up alarms
- Enable CloudTrail logging and analyze user activity
- Set up an Auto Scaling Group with Load Balancer for high availability

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply AWS services such as EC2, IAM, and S3 to deploy virtual machines and manage cloud-based resources.	PO1, PO2, PO5, PO6, PSO1
CO2	Analyze and implement containerization using Docker and integrate it with orchestration tools to manage scalable applications.	PO2, PO3, PO5, PO6, PSO1
CO3	Evaluate different CI/CD tools like Jenkins, GitHub Actions, and AWS CodePipeline to automate software deployment and improve development efficiency.	PO3, PO4, PO5, PSO1
CO4	Create complete DevOps workflows by integrating cloud storage, monitoring tools, version control, and deployment strategies for real-time applications.	PO4, PO5, PO12, PSO1

TEXT BOOK:

1. Gene Kim, Jez Humble, Patrick Debois, John Willis and Nicole Forsgren, “The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations”, IT Revolution Press, 2016.
2. Nigel Poulton, “Docker Deep Dive”, 3rd edition, Leanpub, 2018.

REFERENCE BOOK:

1. Michael Wittig and Andreas Wittig, “Amazon Web Services in Action”, 2nd edition, Manning Publications, 2018.
2. Kelsey Hightower, Brendan Burns and Joe Beda, “Kubernetes Up & Running: Dive into the Future of Infrastructure”, 2nd edition, O’Reilly Media, 2019.
3. Kief Morris, “Infrastructure as Code: Managing Servers in the Cloud”, 2nd edition, O’Reilly Media, 2020.

22IT967 CLOUD FULL STACK DEVELOPMENT

Hours Per Week :

L	T	P	C
-	4	4	4

PREREQUISITE KNOWLEDGE: Web Technologies

COURSE DESCRIPTION AND OBJECTIVE: This course introduces students to provide learners with the essential tools, proven practices, and a collaborative mind set required to unify and streamline software development and IT operations. Covering the entire DevOps lifecycle—from version control and continuous integration/deployment (CI/CD) to infrastructure as code and automated monitoring.

MODULE – 1

UNIT-1:

Introduction to Cloud Architectures

Fundamentals of Cloud Computing, Hybrid Cloud Concepts, Multi-Cloud Concepts, Cloud Service Providers.

UNIT- 2:

Design and Implementation

Architecture Patterns, Networking in Hybrid and Multi-Cloud, Data Management and Migration, Security Considerations(IAM)

PRACTICES:

- Set up a virtual machine using any free-tier cloud provider.
- Classify and compare services as IaaS, PaaS, SaaS using real examples (Google Docs vs Heroku vs EC2).
- Deploy a static website using object storage (e.g., Amazon S3, Azure Blob, or Google Cloud Storage).
- Simulate a hybrid setup by running a local database and connecting it to a web app hosted on the cloud
- Build a simple micro services-based app (e.g., User service + Product service) using Docker and Express.js.
- Simulate an event-driven system using AWS SNS/SQS or Firebase Cloud Messaging to trigger functions between services.
- Simulate hybrid networking by connecting a local virtual machine (e.g., Virtual Box) to a cloud server using ngrok or ZeroTier.

MODULE-2

UNIT-1:

Management and Governance

Monitoring and Logging-Log Aggregation and Analysis, Cost Optimization, Automation and Orchestration- Cost Management Tools and Techniques, Governance and Compliance

UNIT 2:
Deployment
Containerization and Orchestration-Container Registry and Management, API Management, CI/CD Pipelines- Testing and Release Strategies.

PRACTICES:

- Backup and restore a local MySQL/PostgreSQL database to a cloud database service (e.g., Amazon RDS / Google Cloud SQL).
- Use AWS CLI or GCP SDK to upload/download large datasets between local and cloud buckets (S3, GCS).
- Set up real-time sync between a local folder and a cloud storage bucket using Rclone or Gsutil.

- Create IAM users and roles in AWS/Azure/GCP with custom policies (e.g., read-only access to storage, admin access to compute).
- Attach IAM policies to a cloud function or server to restrict access to resources (e.g., allow read-only access to S3 from a Lambda function).

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply modern architecture patterns such as micro services, serverless, and event-driven models in cloud applications.	PO1, PO2, PO5, PSO1, PSO2, PSO3
CO2	Design and simulate networking for hybrid and multi-cloud environments, understanding concepts like VPNs, peering, and load balancing.	PO1, PO2, PO3, PO8, PSO1, PSO2, PSO3
CO3	Implement Identity and Access Management (IAM) principles by configuring roles, permissions, and policies in cloud platforms	PO4, PO5, PO6, PO8, PSO1, PSO2, PSO3
CO4	Develop and deploy secure, scalable, and distributed applications that follow cloud-native principles.	PO3, PO4, PO6, PO12, PSO1, PSO2, PSO3

TEXT BOOK:

1. Concepts, Technology & Architecture, Thomas Erl, Prentice Hall.
2. Principles and Paradigms, Rajkumar Buyya, James Broberg, Andrzej Goscinski, Wiley

REFERENCE BOOK:

1. Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS), Michael J. Kavis, Wiley.
2. How Google Runs Production Systems, Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy, O'Reilly

B.Tech.

OPEN ELECTIVES

COURSE CONTENTS

22IT852 OBJECT ORIENTED PROGRAMMING

Source of Image: https://www.datasciencecentral.com/what-should-java-developers-learn-in-2021/

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Computer Programming

COURSE DESCRIPTION AND OBJECTIVES:

This course is about the fundamentals of Object-Oriented Programming (OOP) Concept and OOP-based software development methodology. Java as a class-based and pure OOP language is used to demonstrate and implement appropriate concepts and techniques. The students are exposed to the concepts, fundamental syntax, and the thought processes behind object oriented programming. By end of the course, students will acquire the basic knowledge and skills necessary to implement object-oriented programming techniques in software development using Java.

MODULE-1

UNIT-1 **16L+0T+8P=24 Hours**

BUILDING BLOCKS OF OOPS:

Introduction: Java buzzwords, OOP principles, Data types, Operators, Control statements, Type conversion and casting, Arrays.

Classes and Methods: Introduction to classes and methods, objects, Constructors, Usage of static, Access control, String class, String Tokenizer.

Inheritance: Basics of Inheritance, Types of inheritance, Abstract classes, Interfaces, Usage of final, creating, defining and accessing Packages

UNIT-2 **8L+0T+8P=16 Hours**

OVERLOADING, INTERFACES AND PACKAGES:

Overloading: Overloading Methods and Constructors, this key word, Usage of super key word, Polymorphism, Method overriding

Interfaces: implementing interface, extending interfaces, accessing a package, importing packages.

PRACTICES:

- Reading different types of data from the user and display that data using Scanner class.
- Illustrating type conversions.
- Implementing different operators.
- Generating electricity bill
- Implementing different patterns.
- Implementing logical programs.
- Implementing Arrays.
- Implementing String class.
- Implementing String Tokenizer class.
- Implementing super keyword.
- Implementing forms of Inheritance
- Implementing overloading and overriding.
- Implementing runtime polymorphism.
- Create an abstract class Media (id, description). Derive classes Book (page count) and CD (play time). Define parameterized constructors. Create one object of Book and CD each and display the details.

SKILLS:

- ✓ *Analyze and develop algorithm for real life problems using Java.*
- ✓ *Able to develop multi-threaded applications.*
- ✓ *Able to create efficient software applications.*

- Define an interface, operations which has method area (), volume (). Define a constant PI having value 3.14. Create class a Cylinder which implements this interface (member-id, height). Create one object and calculate area and volume.
- Implementing packages.

MODULE-2

UNIT-1 **14L+0T+8P=22 Hours**

EXCEPTION HANDLING AND MULTITHREADING:

Exception Handling: Concepts of exception handling, Types of exceptions, Built-in exceptions, Usage of try, catch, throw, throws and finally keywords.

Multithreading: Concepts of Thread, Thread priorities, multithreading, Daemon thread, Synchronization.

UNIT-2 **10L+0T+8P=18 Hours**

AWT, APPLETS AND GUI PROGRAMMING WITH SWING:

AWT and Applets: Concepts of applets, differences between applets and applications, life cycle of applet, AWT, AWT Hierarchy, AWT Controls.

Exploring Swing Controls: JLabel JText Field, JButton, JCheckBox, JRadioButton, JTabbed Pane, JList, JCombo Box.

PRACTICES:

- Implementing Exception handling.
- Implement java program which accepts withdraw amount from the user and throws an exception “In Sufficient Funds” when withdraw amount more than available amount.
- Creating Thread.
- Implementing multithreading.
- Create three threads and that displays “good morning”, for every one second ,”hello” for every 2 seconds and “welcome” for every 3 seconds by using extending Thread class.
- Creating simple Applet.
- Develop an Applet program to accept two numbers from user and output the sum, difference in the respective text boxes.
- Implementing JLabel and JText Field.
- Implementing JButton and JRadioButton.
- Design student registration form using Swing Controls. The form which having the following fields and button SAVE a. Form Fields are: Name, RNO, Mailid, Gender, Branch, Address.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply object-oriented programming principles to develop modular Java programs using classes, objects, methods, and control structures.	PO1, PO2, PO3, PO4, PO6, PO8, PO12, PSO1, PSO3
CO2	Analyze the behavior of Java inheritance, interfaces, packages, and exception handling mechanisms for effective code reuse and robustness.	PO1, PO2, PO3, PO4, PO6, PO8, PO12, PSO1, PSO3
CO3	Evaluate multithreading and collection frameworks to enhance performance, synchronization, and data management in Java applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PSO12, PSO1, PSO3
CO4	Create interactive GUI applications using Swing components and event-driven programming concepts in Java.	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3

TEXT BOOKS:

1. Herbert Scheldt, “Java the complete reference”, 12th Edition, McGraw Hill, Education, 2021.
2. T. Budd, “Understanding Object-Oriented Programming with Java”, Updated Edition, Pearson Education, 2000.

REFERENCE BOOKS:

1. J. Nino and F.A. Hosch, “An Introduction to programming and OO design using Java”, 3rd Edition, John Wiley & sons, 2008.
2. P. Radha Krishna, “Object Oriented Programming through Java”, 1st Edition, Universities Press, 2007.
3. R. A. Johnson, “Java Programming and Object Oriented Application Development”, 1st Edition, Cengage Learning, 2006.

22IT854 PYTHON PROGRAMMING

Image source: https://webnotics.solutions/python-programming/

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Computer Programming

COURSE DESCRIPTION AND OBJECTIVES:

This course offers sufficient knowledge required to understand the fundamental concepts of Python programming language. This course enables students to choose appropriate data structures like lists, dictionaries, tuples, sets, strings for the given problem. In addition, the students will be able to create reliable, modular and reusable applications using Object- Oriented Programming approaches.

MODULE-1

UNIT-1 **6L+0T+6P=12 Hours**

PYTHON BASICS:

Python Installation and Working of it, get familiar with python variables and data types, Operator understanding and its usage, detail study of python blocks, Hands on with conditional blocks using if, else and elif

UNIT-2 **10L+0T+10P=20 Hours**

WORKING WITH VARIETIES OF DATA:

Hands on string handling and looping with range, list, Tuples, Sets and dictionaries. hands on to organize python code with function, modular approach in python.

PRACTICES:

- Installation of python and relevant packages in windows.
- Installation of python and relevant packages in Linux.
- Practice Execution of python statements in REPL(shell).
- Implement a python program to display all the python keywords and display each of them in separate lines.
- Develop a python program to read two integers and perform all possible arithmetic operations on those two numbers.
- Develop a program to accept three numbers as command line arguments and find biggest, smallest and average of those three numbers.
- Implement a python program to find first n Prime Numbers.
- Implement a program that prints the decimal equivalents of 1/2, 1/3, 1/4, . . . ,1/n.
- Implement a python program to read n and find sum of even and odd numbers.
- Write python code to achieve the following
  - o to remove vowels in the given string using control transfer statements.
  - o to count number of uppercase and lowercase letters in the given string.
  - o to remove all punctuation characters from given string.
- Implement python code to illustrate the following on Lists and Tuples
  - o Creation b) Accessing elements c) apply operators d) Usage of different methods
- Implement python code to illustrate the following on Sets and Dictionary
  - o Creation b) Accessing elements c) apply operators d) Usage of different methods
- a) Implement python code to illustrate the following

- Positional arguments ii) Keyword arguments iii) Default arguments iv) Variable length arguments
- Implement a function to find nth Fibonacci number.
- Develop a recursive function to find the factorial of a given number.
- Implement function to compute GCD, LCM of two numbers (use Lambda function).

MODULE-2

UNIT-1 **8L+0T+8P=16 Hours**

EXCEPTION AND FILE HANDLING:

Handling and helping file operations, coding with the exceptional handling

UNIT-2 **8L+0T+8P=16 Hours**

OBJECT ORIENTED PROGRAMMING:

Object-Oriented Programming, Classes and working with instances, Method overloading, Polymorphism

PRACTICES:

- Develop a python code to handle the following built-in exceptions
  - o ValueError ii) ZeroDivisionError iii) TypeError iv) NameErrror
- Implement python code to handle multiple exceptions.
- Implement Python code to raise an exception.
- Implement python code to read contents of a file and write the contents to another file.
- Create a class called Student and perform operations such as display, Calculate percentage, add, delete and modify student data.
- Design python code to depict the following oops concepts: i) Datahiding ii) Inheritance iii) Overriding
- Develop python code to calculate the following statistical parameters using python 'numpy'.
- Mean b) Hormonic Mean c) Meadian d) Mode e) Standard Deviation f) Variance g) Percentile
- Design python code to illustrate the following plots using 'matplotlib' package
- Line plot b) Bar plot c) Histogram d) Scatter Plot
- Implement python program for the following problems on Pandas DataFrame
- Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels. Sample Python dictionary data and list labels:
- exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
  'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
- 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
  'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
- labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
- Write a Pandas program to select the 'name' and 'score' columns from the following DataFrame.
- Write a Pandas program to select the specified columns and rows from a given data frame.
- Write a Pandas program to select the rows where the number of attempts in the examination is greater than 2.
- Write a Pandas program to count the number of rows and columns of a DataFrame.
- Write a Pandas program to change the name 'James' to 'Adhvik' in name column of the DataFrame.

SKILLS:

- ✓ *Identify suitable data types and data structures required for an application*
- ✓ *Design structured and Object oriented programming solutions*
- ✓ *Design reliable applications for a given problem.*

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply Python control structures, and data types to develop basic programs and solve simple computational problems.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO11, PO12, PSO1,PSO3
CO2	Analyze built-in data structures, functions, and file handling features to organize and manipulate data effectively.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO11,PO12, PSO1, PSO3
CO3	Evaluate exception handling, modular programming, and object-oriented principles to enhance the robustness, readability, and maintainability of Python applications.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO11, PO12,PSO1, PSO3
CO4	Create object-oriented Python applications by designing classes, using inheritance, and implementing data encapsulation.	PO1, PO2, PO3, PO4, PO5, PO9, PO10, PO11,PO12, PSO1, PSO3

TEXT BOOK:

1. Reema Thareja, “Python Programming: Using Problem Solving Approach”, 1st Edition, Oxford University Press, 2017.

REFERENCE BOOKS:

1. Eric Matthes, “Python Crash Course: A Hands-On, Project-Based Introduction to Programming”, 2ndEdition, No Starch Press, 2019.
2. John V. Guttag, “Introduction to Computation and Programming Using Python”, 3rd Edition, The MIT Press, 2021
3. Allen B. Downey,” Think Python”, 2nd edition, O’rielly publishing, 2015.
4. Vamsi Kurama, “Python Programming: A Modern Approach”, 1st Edition, Pearson Publishers, 2018.

Image Source: https://honestproscons.com/wp-content/uploads/2021/03/web-technology.jpg

22IT855 WEB TECHNOLOGIES

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: Computer Programming

COURSE DESCRIPTION AND OBJECTIVES:

This course offers the basic concepts used to develop static web pages and it also provides knowledge of Internet programming concepts. Further, this course is to build web applications using HTML, CSS, and client side script technologies that span multiple domains.

MODULE-1

UNIT-1 **4L+8T+8P=20Hours**

HTML BASICS:

Coding Basics: HTML Syntax, html, head, title, and body tags, Headings, paragraphs and lists, The strong and em tags, The doctype, The lang attribute, The meta tag, and the Unicode character set.

Coding Links: Absolute & Relative URLs: Anchor tags and hrefs, Linking to other websites, Linking to pages within a website, Opening a link in a new browser window/tab.

Adding Images: The break tag, The image tag, and source attribute, Using the width, height, and alt attributes, Using horizontal rules, tables, forms, and frames.

UNIT-2 **4L+8T+8P=20 Hours**

CASCADING STYLE SHEETS AND DIV TAGS:

Cascading Style Sheets (CSS): The style tag, Tag selectors, font size, font family, color, & line-height properties, and Hexadecimal color codes.

CSS Properties: Text, background, border, list and font.

CSS Class Selectors: The class attribute, CSS class selectors, The span tag, CSS opacity.

Div Tags, ID Selectors, & Basic Page Formatting: Dividing up content with the div tag, Assigning IDs to divs, Setting width & max-width, CSS background color, Adding padding inside a div, Centering content, CSS borders, CSS shorthand & the DRY principle.

PRACTICES:

- Practice Basic HTML tags
- Create links on same page and other pages
- Insert images on a web page
- Create lists on a web page
- Create Tables on a web page
- Create forms such as login form and registration form etc.
- Working with Frames
- Add different types of CSS to web pages
- Usage of div tag in the web page
- Create a personal website using HTML and CSS.

MODULE-2

UNIT-1 **4L+8T+8P=20 Hours**

JAVA SCRIPT:

Java Script: Introduction, Document Object Model, Language Syntax, Variable declaration, Operators, Control Statements, Understanding Arrays, Function Declaration.

Built-in Functions: Standard Date and Time, String, Array and Math.

UNIT-2 **4L+8T+8P=20 Hours**

HTML FORM VALIDATIONS:

HTML Form Validations: HTML Document Object Model, Working with HTML form and its elements.

Working with Objects and Classes: Working with Objects, Call method in JavaScript, Inheritance in JavaScript using prototype.

Java script Events: Keyboard events, mouse events, form events.

PRACTICES:

- Practice basic JavaScript programs such as the variable declaration and operators.
- Usage of Control Statements in JavaScript.
- Creating and accessing arrays in JavaScript.
- Working with functions in JavaScript.
- Perform validations on HTML forms using JavaScript.
- Working with Cookies.
- Create JavaScript Objects and Classes.
- Apply JavaScript on HTML and CSS webpages.

COURSE OUTCOMES:

Upon successful completion of this course, students will have to ability to:

CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply HTML syntax, tags, attributes, links, images, and tables to design basic web pages with structured content.	PO1, PO2, PO3, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO2	Analyze the use of CSS properties, class and ID selectors, and div tags to control page layout, styling, and responsiveness.	PO1, PO2, PO3, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO3	Evaluate the logic and flow of JavaScript programs using variables, arrays, functions, events, and built-in methods for web interactivity.	PO1, PO2, PO3, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3
CO4	Create dynamic and user-friendly web pages with form validations, DOM manipulation, and object-based JavaScript event handling.	PO1, PO2, PO3, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO1, PSO3

TEXT BOOKS:

1. Jon Duckett, "Beginning Web Programming with HTML, XHTML, and CSS", 2nd Edition, Wiley India Pvt. Ltd, 2010.
2. Julie C. Meloni , "HTML, CSS, and JavaScript All in One", Sams Teach Yourself, 3rd Edition, Pearson, 2015.

REFERENCE BOOKS:

1. Chris Bates, "Web Programming, Building Internet Applications", 3rd Edition, Wiley Dream Tech, 2012.
2. Jon Duckett, "HTML & CSS: Design and Build Websites", 1st Edition, John Wiley & Sons, 2011.

SKILLS:

- ✓ *Apply the CSS and JavaScript on HTML web pages.*
- ✓ *Develop static web pages.*
- ✓ *Create dynamic websites.*

22IT853 OPEN SOURCE WEB TECHNOLOGIES

IMAGE SOURCE: https://thumbs.dreamstime.com/z/painted-hand-shows-concept-hologram-open-source-his-hand-painted-hand-shows-concept-hologram-open-source-his-hand-drawn-man-115398454.jpg

Hours Per Week :

L	T	P	C
2	2	-	3

PREREQUISITE KNOWLEDGE: HTML, CSS and JavaScript

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the concepts pertaining to open-source technologies such as LINUX, MySQL, PHP, Apache web server, and various other tools used to develop web applications. In addition, this course is to offer insight into various open-source technologies to develop web applications.

MODULE-1

UNIT-1 **10L+0T+8P=18Hours**

OPEN SOURCE:

Open Source: Introduction, Open source operating System, Nature of open sources, Advantages, Application of open sources. Introduction to dynamic web content, Setting up a development server, Introduction to PHP, PHP functions and Objects, PHP arrays, Practical PHP.

UNIT-2 **14L+0T+8P=22 Hours**

OPEN SOURCE DATABASE:

Open Source Database: Introduction to MySQL, Accessing MySQL using PHP, querying a MySQL database with PHP, Practical MySQL, preventing hacking attempts, Using MySQL procedure.

Form Handling: Form Handling building forms, Retrieving submitted data, An example program, Cookies, Sessions, and authentication using cookies in PHP, HTTP authentication, Using sessions.

PRACTICES:

- Develop dynamic web content pages.
- Setting up a development server.
- Practice Basic PHP programs.
- Working with PHP functions and Objects.
- Create arrays using PHP.
- Create dynamic web pages using PHP.
- Practice basic MySQL queries.
- Access MySQL database using PHP.
- Create Database Connectivity with PHP & MySQL.
- Working with MySQL procedures.
- Create forms using PHP and retrieve the data from the database.
- Work with Cookies and Sessions in PHP.

MODULE-2

UNIT-1 **10L+0T+8P=18 Hours**

HTML5:

Introduction: to HTML5, The HTML5 Canvas, HTML5 audio, and video, Bringing it all together by designing a social networking site.

SKILLS:

- ✓ *Use the Open source technologies.*
- ✓ *Develop dynamic web pages.*
- ✓ *Design a social networking website.*
- ✓ *Apply the AngularJS concepts to dynamic websites.*
- ✓ *Create animations using AngularJS.*

UNIT-2 **14L+0T+8P=22 Hours**

ANGULAR JS:

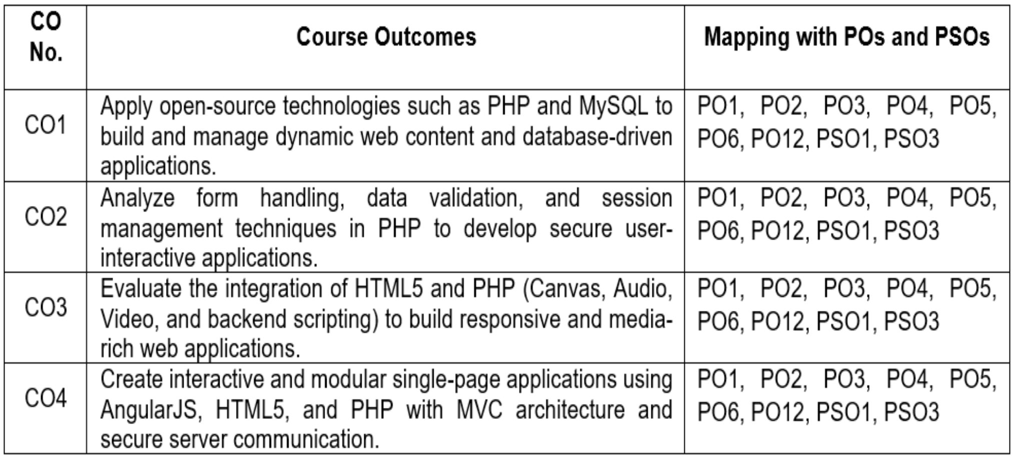
Angular JS: The basics of AngularJS, Introduction MVC, Filters, and modules, Directives, Working with Forms, Services and server communication, Organizing views, Angular JS animation.

PRACTICES:

- Practice basic HTML5 tags.
- Usage of HTML5 audio, video, and canvas tags.
- Build a Social Networking Website.
- Practice basic AngularJS.
- Working with forms, services, and server communication through AngularJS.
- Using views in AngularJS applications.
- Create animations using AngularJS.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:



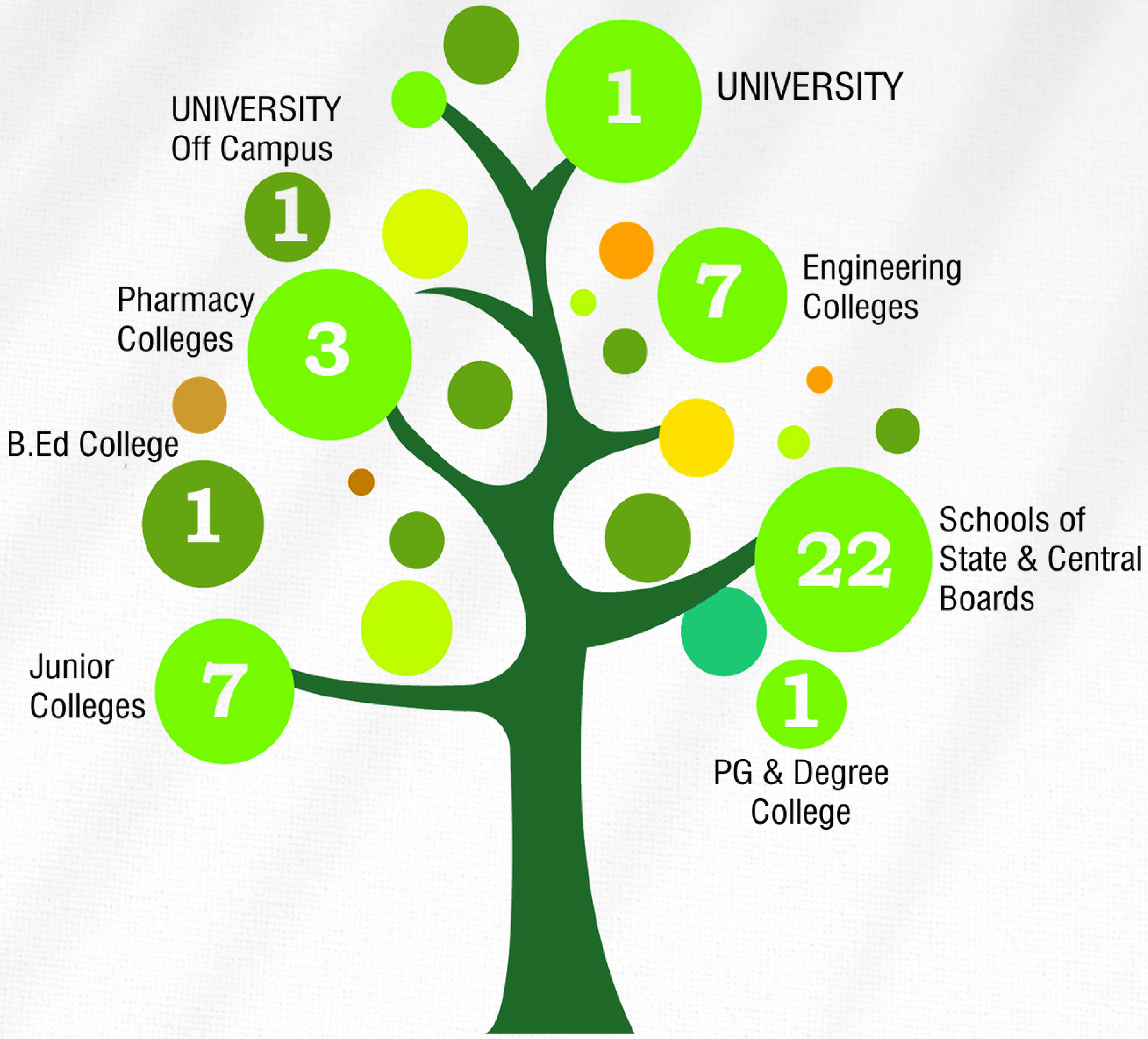
CO No.	Course Outcomes	Mapping with POs and PSOs
CO1	Apply open-source technologies such as PHP and MySQL to build and manage dynamic web content and database-driven applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3
CO2	Analyze form handling, data validation, and session management techniques in PHP to develop secure user-interactive applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3
CO3	Evaluate the integration of HTML5 and PHP (Canvas, Audio, Video, and backend scripting) to build responsive and media-rich web applications.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3
CO4	Create interactive and modular single-page applications using AngularJS, HTML5, and PHP with MVC architecture and secure server communication.	PO1, PO2, PO3, PO4, PO5, PO6, PO12, PSO1, PSO3

TEXT BOOKS:

1. Godfrey Boyle, “Renewable Energy, Power for a Sustainable Future”, 1st edition, Oxford URobin Nixon, “Learning PHP, MySQL & JavaScript WITH JQUERY, CSS & HTML5”, 5th Edition, O’Reilly, 2018.
2. Andrew Grant, “Beginning Angular JS”, 1st Edition, Apress, 2014.

REFERENCE BOOKS:

1. Steve Prettyman, “Learn PHP 7 Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL”, 1st Edition, Apress, 2015.
2. Adrian W. West and Steve Prettyman, “Practical PHP 7, MySQL 8, and MariaDB Website Databases: A Simplified Approach to Developing Database-Driven Websites”, 2nd Edition, A Press, 2018.



1 UNIVERSITY

1 UNIVERSITY Off Campus

7 Engineering Colleges

3 Pharmacy Colleges

1 B.Ed College

22 Schools of State & Central Boards

7 Junior Colleges

1 PG & Degree College

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be UNIVERSITY)
-Estd. u/s 3 of UGC Act 1956

Vadlamudi, Guntur Dist. - 522 213 A.P. India
Phone: 0863 2344700 | www.vignan.ac.in