B.Tech. COMPUTER SCIENCE AND ENGINEERING

TABLE OF CONTENTS

FOREWORD

The vital aspect of Computer Science & Engineering program is problem solving through programming. It is an essential skill required to carry out profession of Software Engineer. CSE graduates practice the activities (analysis, design, development, testing and maintenance) of software development life cycle to solve complex and challenging real world problems.

R22 Curriculum enables our students to start with the basic science, basic engineering and introductory programming courses. Further, they learn the mathematical foundations of computing; get hands-on experience in building software solutions using various technologies for various real-world problems and pursue advanced courses such as Artificial Intelligence, Machine Learning, Cloud Computing, Big Data & Analytics, Internet of Things and much more.

Salient Features of R 22 Curriculum:

- *Multidisciplinary holistic education with continuous learning and continuous assessment.*
- *Lateral entry and lateral exit options.*
- *Credit Earning by credit transfer.*
- *Honors/ Research Honors/ Minor/ Add-on Diploma/ Add-on Certification and Dual B.Tech.+ M.Tech./ MBA Degree of 5 Years.*
- *Semester drop option to pursue innovation, incubation, entrepreneurial and advanced exploratory activities and subsequent re-entry.*

Emphasis on continuous formative assessment with a creative summative assessment will facilitate the student to "Move away from high stake examinations – towards more continuous and comprehensive evaluation".

The Board of Studies of B.Tech. CSE Programme consists of a right mix of eminent personalities from Academic, Research and Industry Organizations, besides experienced faculty members of the University.

External BoS Members:

1. Prof. C. R. Rao, Professor SCIS, University of Hyderabad.
2. Prof. R. B. V. Subramanyam, Professor, Department of CSE, NIT Warangal.
3. Dr. B. Venkata Ramana, Assoc. Professor & HoD, Department of CSE, IIT Tirupathi.
4. Dr. Nagesh Bhattu Sristy, Asst. Professor, Department of CSE, NIT AP.

I thank all the BOS members, Academic Council Members and University authorities for their continuous support and encouragement towards design of this innovative curriculum for CSE.

Dr. K.V. Krishna Kishore
Head, Department of CSE
VFSTR Deemed to be University

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 COMPUTER SCIENCE & ENGINEERING

VISION of the department

To evolve as a centre of high repute in Computer Science & Engineering and create computer software professionals trained on problem solving skills imbued with ethics to serve the ever evolving and emerging requirements of IT Industry and society at large.

MISSION of the department

M_1: Imparting quality education through well designed curriculum, innovative teaching and learning methodologies integrated with professional skill development activities to meet the challenges in the career.

M_2: Nurture research and consultancy activities amongst students and faculty by providing State-of-art facilities and Industry-Institute Interaction.

M_3: Developing capacity to learn new technologies and apply to solve social and industrial problems to become an entrepreneur.

B.Tech in Computer Science and Engineering

Program Educational Objectives (PEOs)

PEO1: Graduates acquire extensive technical knowledge and related skills required to demonstrate themselves as professionals or pursue higher education.

PEO2: Graduates adapt to upskilling and excel in their careers despite future technological changes, and demonstrate research aptitude to generate innovative engineering solutions.

PEO3: Graduates acquire the potential to contribute for the field of computing as well as for the societal development by demonstrating professional, social and ethical practices.

Program Specific Outcomes (PSOs)

The students will be able to

PSO1: **Application Development Skills:** Design and development of web applications using various technologies such as HTML, JSP, PHP, ASP and ASP.NET to cater the needs of the society.

PSO2: **Enrich Research Skills:** Offer solutions which impact geo-socio-economic and environmental scenarios by using Machine Learning, Artificial Intelligence and IoT.

Program Outcomes (POs)

PO1: Engineering knowledge

PO2: Problem analysis

PO3: Design/development of solutions

PO4: Conduct investigations of complex Problems

PO5: Modern tool usage

PO6: The engineer and society

PO7: Environment and sustainability

PO8: Ethics

PO9: Individual and team-work

PO10: Communication

PO11: Project management and finance

PO12: Life-long learning

Student Outcomes (SOs) - Engineering Accreditation Commission (EAC)

SO1: An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.

SO2: An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as a global, cultural, social, environmental, and economic factors.

SO3: An ability to communicate effectively with a range of audiences.

SO4: An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.

SO5: An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.

SO6: An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

SO7: An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

Student Outcomes (SOs) - Computing Accreditation Commission (CAC)

SO1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

SO2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

SO3: Communicate effectively in a variety of professional contexts.

SO4: Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

SO5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.

SO6: Apply computer science theory and software development fundamentals to produce computing-based solutions.

R22 B.Tech.
4 YEAR
DEGREE PROGRAMME

COURSE STRUCTURE - R22

I Year I Semester

Course Code	Course Title	L	T	P	C
22MT103	Linear Algebra and Ordinary Differential Equations	3	2	0	4
22PY105	Semiconductor Physics and Electromagnetics	3	0	2	4
22EE101	Basics of Electrical and Electronics Engineering	2	0	2	3
22CT103	Engineering Chemistry	3	0	2	4
22TP105	Problem Solving through Programming - I	2	2	2	4
22EN102	English Proficiency and Communication Skills	0	0	2	1
22TP101	Constitution of India	0	2	0	1
22SA101	Physical Fitness, Sports and Games-I	0	0	3	1
Total		**13**	**6**	**13**	**22**
		32 Hrs			**22**

I Year II Semester

Course Code	Course Title	L	T	P	C
22MT106	Algebra	3	2	0	4
22MT107	Discrete Mathematical Structures	3	2	0	4
22ME101	Engineering Graphics	2	0	2	3
22TP106	Problem Solving through Programming - II	1	2	2	3
22EN104	Technical English Communication	2	0	2	3
22MT108	Numerical Methods	3	2	0	4
22SA102	Orientation Session	0	0	6	3
22SA103	Physical Fitness, Sports and Games – II	0	0	3	1
Total		**14**	**8**	**15**	**25**
		37 Hrs			**25**

COURSE STRUCTURE - R22

R22 B.Tech. 4 YEAR DEGREE PROGRAMME

II Year I Semester

Course Code	Course Title	L	T	P	C
22ST202	Probability and Statistics	3	0	2	4
22TP201	Data Structures	2	2	2	4
22MS201	Management Science	2	2	0	3
22CS201	Database Management Systems	2	2	2	4
22CS202	Digital Logic Design	2	0	2	3
22CS203	Object-Oriented Programming through JAVA	2	0	2	3
22CT201	Environmental Studies	2	1	0	2
22SA201	Life Skills-I	0	0	2	1
	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication.	0	0	0	1
Total		**15**	**7**	**12**	**25**
		34 Hrs			**25**

II Year II Semester

Course Code	Course Title	L	T	P	C
22TP203	Advanced Coding Competency	0	0	2	1
22TP204	Professional Communication	0	0	2	1
22CS205	Computer Organization and Architecture	2	2	0	3
22CS206	Design and Analysis of Algorithms	2	2	2	4
22CS207	Operating Systems	2	0	2	3
22CS208	Theory of Computation	3	2	0	4
22SA202	Life Skills - II	0	0	2	1
	Open Elective – 1	3	0	0	3
	Total	**12**	**6**	**10**	**20**
	Minor / Honours - 1	3	0	2	4
Total		**15**	**6**	**12**	**24**
		33 Hrs			**24**

R22 B.Tech.
4 YEAR
DEGREE PROGRAMME

COURSE STRUCTURE - R22

III Year I Semester

Course Code	Course Title	L	T	P	C
22TP301	Soft Skills Laboratory	0	0	2	1
22CS301	Introduction to Artificial Intelligence	2	2	0	3
22CS302	Compiler Design	2	2	0	3
22CS303	Web Technologies	2	0	2	3
22CS304	Inter-Disciplinary Project – Phase I	0	0	2	0
22CS305	Industry Interface Course	1	0	0	1
	Department Elective – 1	3	0	2	4
	Open Elective – 2	3	0	0	3
	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication.	0	0	0	1
Total		**13**	**4**	**8**	**19**
	Minor / Honours - 2	3	0	2	4
Total		**16**	**4**	**10**	**23**
		30 Hrs			**23**

III Year II Semester

Course Code	Course Title	L	T	P	C
22TP302	Quantitative Aptitude and Logical Reasoning	1	2	0	2
22CS204	Computer Networks	3	0	2	4
22CS306	Data Mining Techniques	2	0	2	3
22CS307	Software Engineering	2	0	2	3
22CS308	Inter-Disciplinary Project – Phase II	0	0	2	2
	Department Elective – 2	3	0	2	4
	Open Elective – 3	3	0	0	3
Total		**14**	**2**	**10**	**21**
	Minor / Honours - 3	3	0	2	4
Total		**17**	**2**	**12**	**25**
		31 Hrs			**25**

COURSE STRUCTURE - R22

R22 B.Tech. 4 YEAR DEGREE PROGRAMME

IV Year I Semester

Course Code	Course Title	L	T	P	C
22CS401	Cryptography and Network Security	2	0	2	3
22CS402	Big Data Analytics	2	0	2	3
22CS403	Cloud Computing	2	0	2	3
	Department Elective – 3	3	0	2	4
	Department Elective – 4	3	0	2	4
	Total	**12**	**0**	**10**	**17**
	Minor / Honours – 4	3	0	2	4
	Total	**15**	**0**	**12**	**21**
		27 Hrs			**21**

IV Year II Semester

Course Code	Course Title	L	T	P	C
22CS404	Project Work	0	2	22	12
	Total	**0**	**2**	**22**	**12**
	Minor / Honors – 5	0	2	6	**4**
	Total	**0**	**4**	**28**	**16**
		32 Hrs			**16**

R22 B.Tech. 4 YEAR DEGREE PROGRAMME

COURSE STRUCTURE - R22

Department Electives

Course Code	Course Title	L	T	P	C
22CS801	Advanced Data Structures	2	2	2	4
22CS802	Advanced JAVA Programming	2	2	2	4
22CS803	Computer Graphics	2	2	2	4
22CS804	Deep Learning	3	0	2	4
22CS805	Digital Forensics	3	0	2	4
22CS806	Digital Image Processing	2	2	2	4
22CS807	Web and Database Security	3	0	2	4
22CS808	Machine Learning	3	0	2	4
22CS809	Mobile Ad-hoc Networks	3	0	2	4
22CS810	Mobile Application Development	2	0	4	4
22CS811	Text Mining	3	0	2	4
22CS812	Numerical Algorithms	3	2	0	4
22CS813	Operating System Design	3	0	2	4
22CS814	Optimization Techniques	3	2	0	4
22CS815	Intrusion Detection and Prevention System	3	0	2	4
22CS816	Simulation and Modeling	3	0	2	4
22CS817	Parallel and Distributed Computing	3	2	0	4

Honours for CSE

Course Code	Course Title	L	T	P	C
22CS951	Advanced Graph Algorithms	3	0	2	4
22CS952	Biometrics	3	0	2	4
22CS953	Parallel and Distributed Computing	3	2	0	4
22CS954	Internet of Things	3	0	2	4
22CS955	Wireless Sensor Networks	3	0	2	4
22CS956	Capstone Project	0	2	6	4

I

YEAR

COMPUTER SCIENCE AND ENGINEERING

B.Tech.

COURSE CONTENTS

I SEM & II SEM

I SEMESTER

22MT103	-	Linear Algebra and Ordinary Differential Equations
22PY105	-	Semiconductor Physics and Electromagnetics
22EE101	-	Basics of Electrical and Electronics Engineering
22CT103	-	Engineering Chemistry
22TP105	-	Problem Solving through Programming - I
22EN102	-	English Proficiency and Communication Skills
22TP101	-	Constitution of India
22SA101	-	Physical Fitness, Sports and Games-I

II SEMESTER

22MT106	-	Algebra
22MT107	-	Discrete Mathematical Structures
22ME101	-	Engineering Graphics
22TP106	-	Problem Solving through Programming - II
22EN104	-	Technical English Communication
22MT108	-	Numerical Methods
22SA102	-	Orientation Session
22SA103	-	Physical Fitness, Sports and Games-II

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	0	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	Blooms Level	Module No.	Mapping with POs
1	Apply the concepts of rank, eigen values and eigenvectors of a matrix and finding inverse of a matrix and powers of a matrix.	Apply	1	1, 2, 9, 10, 12
2	Apply differential equations in real life problems.	Apply	2	1, 2, 9, 10, 12
3	Analyse the solution of a system of linear equations and find it.	Analyze	1	1, 2, 9, 10, 12
4	Inspect the analytical method for solving differential equations and applications.	Analyze	2	1, 2, 9, 10, 12

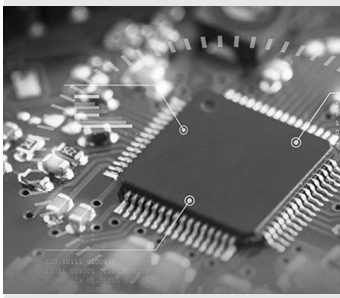
TEXT BOOKS:

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", 44th edition, Khanna Publishers, 2018.

REFERENCE BOOKS:

1. Erwin Kreyszig, "Advanced Engineering Mathematics", 10th Edition, John Wiley & Sons, Inc, 2015.
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, 2020.
4. T. K.V. Iyengar et al, "Engineering Mathematics, I, II, III", S. Chand & Co., New Delhi.

22PY105 SEMICONDUCTOR PHYSICS AND ELECTROMAGNETICS



Source: https://www.scitusacademics.com/product/semiconductor-devices-and-circuits/

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Basics of vectors and semiconductors.

COURSE DESCRIPTION AND OBJECTIVES:

This course ensures commensurable understanding of electrostatics and magnetostatics. 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 **12L+0T+10P=22 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 **12L+0T+6P=18 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.
- Sonometer- Determination of AC frequency.

MODULE-2

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

SEMICONDUCTOR 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 **12L+0T+8P=20 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 fiber - Determination of Numerical aperture – Acceptance angle.
- Determination of Energy Band gap of p-n junction diode.
- Hall Effect - Determination of Hall coefficient.
- Solar cell – Determination of Fill factor & efficiency.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply Maxwell's equations to unravel electron dynamics in amidst of electric and magnetic fields.	Apply	1	1, 2, 4, 5, 9, 10
2	Discriminate solids based on principles of quantum mechanics.	Analyse	1	1, 2, 3, 4, 9, 10
3	Assessment of semiconductors in the perspective of optoelectronic devices.	Evaluate	2	1, 3, 4, 5, 6, 9, 10
4	Comprehend the knowledge of Lasers and optical fibers to conceive their applications in vivid domains.	Apply	2	1, 2, 3, 5, 9, 10

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.

REFERENCE BOOKS:

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	0	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 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 this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze the resistive circuits with independent sources and find its solution.	Analyze	1,2	1,2,6,9
2	Solve the AC (single and three phase) and DC circuits using different methods.	Apply	1,2	1,2,9,12
3	Apply the concepts of electromagnetism for its applications.	Apply	2	1,2,3,9,12
4	Examine the different electrical equipment.	Evaluate	2	1,2,9,12
5	Acquire the knowledge of semiconductor devices to create circuits.	Create	1	1,2,3,9,12

TEXT BOOKS:

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

22CT103 ENGINEERING CHEMISTRY

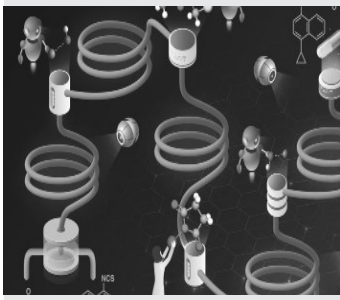


Image source: https://www.rsc.org/journalsbooks-databases/about-journals/reaction-chemistryengineering/

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Concept of bonding, chemical reactions and electrochemical cell

COURSE DESCRIPTION AND OBJECTIVES:

The course aims to cover the importance of chemistry and its applications in engineering disciplines particularly focusing on developing new engineering materials (such as polymers, nanomaterials etc.) and understanding their property for scientific and engineering applications. In addition, the students are also expected to acquire knowledge on electrochemistry and construction of batteries and fuel cells.

MODULE-1

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

POLYMERS:

Introduction, classification, molecular weight determination, (Mw & Mn), types of polymerization, preparation, properties and applications of PE, PMA, Nylon-6,6; Rubber-vulcanization, synthetic rubbers – Neoprene, Introduction to polymer composites, glass fiber and metal oxide/metal composites.

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

NANOMATERIALS & ENGINEERING MATERIALS:

Nanomaterials: Introduction, classification, properties, Top-down (Ball Milling) and Bottom-up (Sol-Gel) synthetic methods; Synthesis, properties and applications of Carbon Nanotubes and Graphene.

Engineering Materials: Lubricants – classification, viscosity, viscosity index, flash and fire points, cloud and pour points and mechanical stability; Refractories – classification, refractoriness RUL, chemical & thermal stability.

PRACTICES:

- Synthesis of Nanoparticles.
- Synthesis of Bakelite.
- Determination of viscosity of oil (Biodiesel, castor oil and coconut oil).
- Water Analysis.
- Synthesis of Iron oxide nanoparticles.
- Synthesis of Au/Ag nanoparticles using plant extract.
- Preparation of Nylon-6,6.
- Preparation of Polystyrene.

MODULE-2

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

ELECTROCHEMISTRY & CORROSIONS:

Electrochemical cells: Galvanic and electrolytic cells; redox reactions; electrode potential; electrochemical series, emf of an electrochemical cell; nernst equation - applications and significances; reference electrodes – standard hydrogen electrode.

SKILLS:

- ✓ *Synthesize various polymers.*
- ✓ *Synthesize nanomaterials.*
- ✓ *Identify the properties of different industrially relevant engineering materials.*
- ✓ *Understand the different components of an electrochemical cell.*
- ✓ *Design electrochemical cell such as battery.*
- ✓ *Identify the types of energy conversion/ storage systems.*

Corrosion: Introduction, dry & wet corrosion; Galvanic series; Corrosion prevention by cathodic protection.

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

BATTERIES & FUEL CELLS:

Batteries: Introduction and importance; Classification of batteries - Lead-acid storage cell and Lithium-ion batteries.

Fuel Cells: Classification of Fuel Cells; Construction, working principle and applications of Hydrogen-Oxygen fuel cell, Biofuel cells - Microbial fuel cells.

PRACTICES:

- Determination of EMF and ΔG of an electrochemical cell.
- Determination of rate of corrosion by weight loss method.
- Construction of Batteries.
- Determination of molecular weight by viscometer.
- Determination of Fe (II) by dichrometry method.
- Determination of available chlorine in bleaching powder.
- Determination of strength of Weak acid by pH-method.
- Determination of concentration of mixture of acids by conductometry.
- Electroplating of Copper and Zinc on metallic objects.
- Determination of rate of corrosion by weight loss method.
- Construction of galvanic cell and measure the emF.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply various synthetic methods for preparing polymers for engineering applications.	Apply	1	1, 2, 4, 9, 10, 11, 12
2	Analyze characteristics in different engineering nanomaterials for the applications of electronic engineering.	Analyze	1	1, 2, 3, 5, 9, 10, 11, 12
3	Distinguish different types of electrochemical cells and corrosions for the real time analysis.	Analyze	2	1, 2, 3, 5, 6, 7, 9, 10, 11, 12
4	Analyze possible corrosion types and their different protection methods.	Analyze	2	1, 2, 3, 5, 6, 7, 9, 10, 11, 12
5	Recommend the principle of electrochemistry for designing various batteries and fuel cells.	Evaluate	2	1, 2, 3, 5, 6, 7, 9, 10, 11, 12

TEXT BOOKS:

1. S. Chawala, “A Textbook of engineering Chemistry engineering materials and Applications”, Dhanpat Rai Publications, 3rd edition, 2015.
2. P.C. Jain and m. Jain, “engineering Chemistry”, Dhanpat Rai Publications, 17th edition, 2015.

REFERENCE BOOKS:

1. K. S. maheswaramma and m. Chugh, “engineering Chemistry”, Pearson, 1st edition, 2015.
2. B. S. Bahl, Arun Bahl and B. D. Tuli, “essentials of Physical Chemistry”, S. Chand and Co. Ltd., 2007.
3. G. Raj and C. Anand, “Instrumental methods of Analysis”, Himalaya Publications, 5th edition, 2007.
4. T. Pradeep, “Nano: The essentials; Understanding of Nano Science and Technology” Tata mcGraw-Hill, New Delhi, 2012.
5. J. mendham, R. C. Denney, J. D. Bares, m. Thomas and B. Siva Sankar, “Vogel’s Textbook of Qualitative Chemical Analysis” (vol. 1), Pearson Publications, 2009

.

Source: Techgig.com

22TP105 PROBLEM SOLVING THROUGH PROGRAMMING - I

Hours Per Week :

L	T	P	C
2	2	2	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+8T+8P=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.

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

CONTROL STATEMENTS

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, continue, goto, and return.

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:

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

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

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

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

3. 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.*

***
****
*****

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

*
**
***
****
*****

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

6. 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

7. 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

8. 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

9. 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

10. 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.

MODULE-2

UNIT-1 **8L+8T+8P=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.

.

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

FUNCTIONS

Functions: 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; Recursion; Library functions.

PRACTICES:

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.

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.

COURSE OUTCOMES:

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

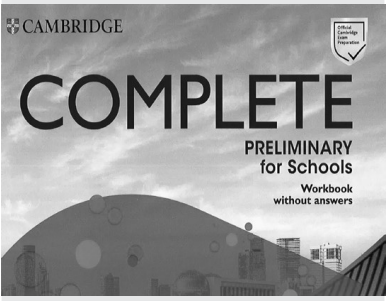
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Identify suitable data type for operands and design of expressions having right precedence.	Apply	1,2	1
2	Apply decision making and iterative features of C Programming language effectively.	Apply	1,2	1
3	Select problem specific data structures and suitable accessing methods	Analyze	1,2	1,2
4	Design and develop non- recursive and recursive functions and their usage to build large modular programs and also able to design string manipulation functions.	Create	1,2	3
5	Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.	Evaluate	1,2	3,4

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. R G Dromey and Pearson, "How to solve it by Computer", 2nd Edition, 1998.



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
0	0	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	Blooms Level	Module No.	Mapping with POs
1	Apply to read and grasp content on a range of topics/texts related to their everyday life like notifications, advertisements, travel brochures, news reports, articles.	Apply	1	7, 8, 9, 10, 12
2	Apply suitable strategies to achieve comprehension, like listening for main points and checking comprehension using contextual clues etc.	Apply	1	7, 8, 9, 10, 12
3	Use functional English to communicate and interact effectively in everyday situations.	Apply	1, 2	7, 8, 9, 10, 12
4	Demonstrate vocabulary beyond that of the familiar subjects.	Analyze	1, 2	7, 8, 9, 10, 12
5	Show sufficient control of English grammar and sentence variety to coherently organise information at sentence and discourse levels.	Evaluate	2	7, 8, 9, 10, 12

TEXT BOOK:

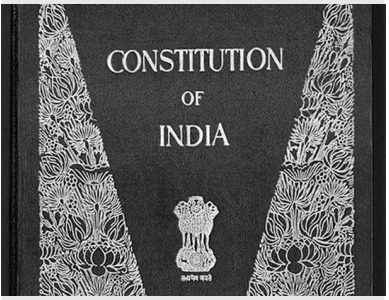
1. Emma Heyderman and Peter May, “Complete Preliminary”, Student’s Book with Answers, 2nd edition, Cambridge University Press, 2019.

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*



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

22TP101 CONSTITUTION OF INDIA

Hours Per Week :

L	T	P	C
0	2	0	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 H 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	Blooms Level	Module No.	Mapping with POs
1	Analyse major articles and provisions of the Indian constitution.	Analyze	1	6
2	Appreciation for the constitution and safeguarding individual rights.	Apply	1	6
3	Evaluating functions of various organs of the State in a democracy.	Evaluate	2	6

TEXT BOOK:

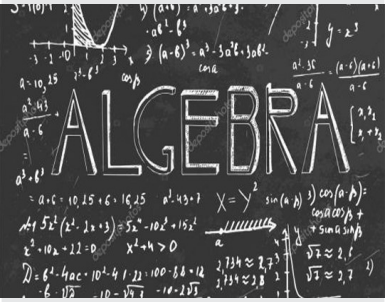
1. PM Bhakshi, “Constitution of India”, 15th edition, Universal Law Publishing, 2018.

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:

- ✓ *Knowledge 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*



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

22MT106 ALGEBRA

Hours Per Week :

L	T	P	C
3	2	0	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**

SKILLS:

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

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.

COURSE OUTCOMES:

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

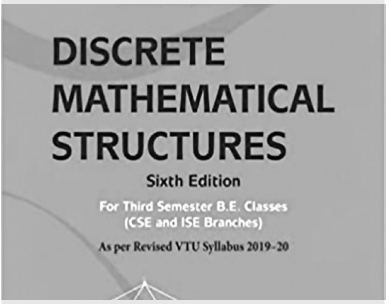
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply the concepts of cosets to study properties of subgroups.	Apply	1	1, 2, 9, 10, 12
2	Outline the various properties and apply group actions critically.	Apply	1	1, 2, 9, 10, 12
3	Understand and apply the concepts of vector spaces, subspaces, bases, dimension and their properties.	Apply	2	1, 2, 9, 10, 12
4	Analyse inner product spaces for their orthogonality.	Analyse	2	1, 2, 9, 10, 12

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



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

22MT107 DISCRETE MATHEMATICAL STRUCTURES

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Basic 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 **12L+8T+0P=20 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 **12L+8T+0P=20 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 **12L+8T+0P=20 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 **12L+8T+0P=20 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	Blooms Level	Module No.	Mapping with POs
1	Apply the concepts of mathematical logic and Boolean algebra.	Apply	1	1, 2, 9, 10, 12
2	Apply Karnaugh map to minimize.	Apply	1	1, 2, 9, 10, 12
3	Solve generating function by recurrence relations.	Apply	2	1, 2, 9, 10, 12
4	Model and solve real world problems using graphs and trees.	Apply	2	1, 2, 9, 10, 12

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

22ME101 ENGINEERING GRAPHICS

Hours Per Week :

L	T	P	C
2	0	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

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

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.

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.

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.*
- ✓ *Know projections by visualization.*

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Communicate the technical ideas in the form of drawings.	Apply	1	1,2,3,5
2	Apply the drawing skills in representing various geometrical features.	Apply	1	1,2,3,5
3	Develop orthographic projections and isometric views of various objects.	Apply	1	1,2,3,5
4	Estimate the lateral surface area of regular geometrical solids.	Analyze	2	1,2,3,5
5	Sketch simple objects and their pictorial views using AutoCAD.	Analyze	2	1,2,3,5

TEXT BOOKS:

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

REFERENCE BOOKS:

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

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

22TP106 PROBLEM SOLVING THROUGH PROGRAMMING - II

Hours Per Week :

L	T	P	C
1	2	2	3

PREREQUISITE KNOWLEDGE: Programming in C.

COURSE DESCRIPTION AND OBJECTIVES:

This course is aimed to impart knowledge on some advanced concepts of C programming language and problem solving. It covers pointers, structures, unions and 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 **4L+8T+8P=20 Hours**

Pointers: Declaration, Initialization, Multiple indirection, Pointer arithmetic, Relationship between arrays and pointers

Problem Solving on Global Coding Platform using Data types and operators; Control statements; functions

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

POINTERS: Scaling up - array of arrays, array of pointers, pointer to a pointer and pointer to an array; Dynamic memory allocation functions.

Problem Solving on Global Coding Platform using Data types and operators; Control statements; functions.

PRACTICES:

1. a. What is a pointer in C, and why is it important in C programming?
   b. A program to declare a pointer, assign a value, and print its value.
2. a. Explain the difference between the address-of operator (&) and the dereference operator (*) in C.
   b. A program to demonstrate pointer arithmetic, e.g., adding to and subtracting from pointers.
3. a. Explain the concept of a double pointer (pointer to a pointer).
   b. A program showing the difference between pointers to constants and constant pointers.
4. a. How can you use pointers to dynamically allocate memory in C?
   b. A program using malloc to allocate memory for an array and then freeing it with free.
5. a. What is the purpose of a null pointer?
   b. A program using null pointers and checking for null before dereferencing.

MODULE-2

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

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

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:

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

2. 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.

3. 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

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.*

The author name is: balaguruswamy

The book ID is: 23413

The book price is: 500.00

4. 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

5. 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.

6. 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.
7. 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.
8. 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

9. 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

10. 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.

11. 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.

12. 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.”

13. 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

14. 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 this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Identify suitable data type for operands and design of expressions having right precedence.	Apply	1,2	1
2	Apply decision making and iterative features of C Programming language effectively.	Apply	1,2	1
3	Select problem specific data structures and suitable accessing methods.	Analyze	1,2	1,2
4	Design and develop non- recursive and recursive functions and their usage to build large modular programs and also able to design string manipulation functions.	Create	1,2	3
5	Develop C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.	Evaluate	1,2	3,4

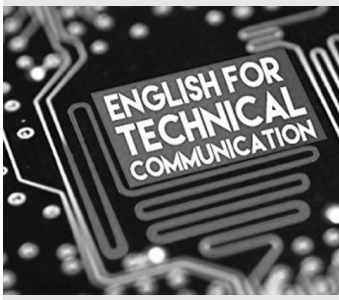
TEXT BOOKS:

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. R G Dromey and Pearson, “How to solve it by Computer”, 2nd Edition, 1998.Ibeaquis alique volorepro omnias dolum et pos es ma que pro doluptae porum inietur, quas minctatibus del molest odit verias imoluptaqui arum cum ius enet alique sitios est poreictint a sunt rent et experum nonsectempos at etus, quat ommos seque in comnihitatis es sequunt quam quo offictis ex eum fuga. Namus erchici endictati int as qui doloriberro odignation re et erferae. Alignatat ut atias culpa voluptatur mil et etum autassi ntotatur?

22EN104 TECHNICAL ENGLISH COMMUNICATION



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

Hours Per Week :

L	T	P	C
2	0	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 **8L+0T+8P=16 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 **8L+0T+8P=16 Hours**

ALIENS

Reading: 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 inVerb 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.

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.*

PRACTICES:

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

MODULE-2

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

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply a variety of strategies to interpret and comprehend spoken texts/ discourse using contextual clues.	Apply	1	6, 7, 8, 9, 10, 12
2	Apply appropriate reading strategies to interpret content / material related to engineering and technology domain.	Apply	1	6, 7, 8, 9, 10, 12
3	Participate in discussions and make short presentations on general and technical topics.	Apply	1, 2	6, 7, 8, 9, 10, 12
4	Possess an ability to write clearly on topics related to technology and workplace communication.	Analyze	2	6, 7, 8, 9, 10, 12
5	Choose functional language, grammar structures, cohesive devices and skills of organisation to express clearly in speaking.	Evaluate	2	6, 7, 8, 9, 10, 12

LANGUAGE LAB ACTIVITIES

Session - 1: Dictionary Skills.

Session - 2: Introduction to Phonetics and Identifying Phonemes.

Session - 3: Pronunciation Practice (Commonly mispronounced words).

Session - 4: Rosetta Stone (Exercises on LSRW).

Session - 5: Listening Comprehension (Summarising exercise on a Ted Talk).

Session - 6: Technical Presentations (Individual).

Session - 7: Technical Presentations (Team).

Session - 8: TOEFL Mastery.

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”, 2ndEdition, McGraw Hill Education, 2017.
5. Narayana Swamy V R, “Strengthen your Writing”, Third Edition Orient Black Swan, New Delhi, 2005.

Source:https://www.bragitoff.com/2015/11/numerical-analysis-c-programs-for-various-techniques/

22MT108 NUMERICAL METHODS

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Advanced calculus, (advanced) linear algebra, and differential equations.

COURSE DESCRIPTION AND OBJECTIVES:

The course is intended to train the students in various numerical methods starting from finding roots of polynomial equations to the extent of solving partial differential equations. Numerical methods, based upon sound computational mathematics, are the basic algorithms underpinning computer predictions in modern systems science. This course includes techniquesfor optimization, interpolation from the known to the unknown, linear algebra underlying systems of equations, ordinary differential equations to simulate systems.

MODULE-1

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

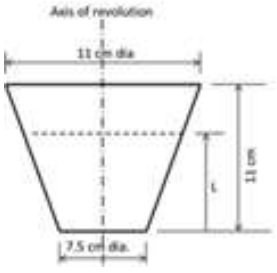
NUMERICAL SOLUTION OF EQUATIONS AND SYSTEM OF EQUATIONS: Solution of Algebraic and Transcendental Equations, bisection, regula falsi, newton's method, Horners method, Graeffe.s root squaring method, Solution of Simultaneous Linear Algebraic Equation, gauss elimination gauss Jordan, method of Triangularization, Crouts method, Jacobi method, Seidal method, Relaxation Method.

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

INTERPOLATION: Finite Differences and properties, Interpolation of equal and Unequal Intervals, forward and backward interpolation, Central Difference Interpolation Formulae (For Equal Intervals), devided difference and lagranges interpolation formula, stirling formula for equal and unequal intervals.

PRACTICES:

- Determine the solution of the nonlinear polynomial Equation f(x): x4-2x3+x2-3x+3=0 using the Horners method, Giraffe's root squaring method.
- Use the Newton-Raphson's method to find the roots of the following nonlinear polynomial equation: f(x) = x4+7x3+x2+5x+3=0
- Determine the locate the mark line for the content volume of 500 milli liter in a measuring cup with its dimensions shown in the right of Figure



- Determine the solution of the following differential equation using the forward finite difference scheme $\frac{d^2x(t)}{dt^2} + x(t) = 0$, with specified initial conditions x(0)=1,x^1 (0)=0.
- Solve Equations 2x+5y=21,x+2y=8,3x+5y=12 by using Crout's method.

SKILLS:

- ✓ *Solving differential equations.*
- ✓ *Computing roots of polynomial equations.*
- ✓ *Interpolate given data.*

MODULE-2

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

NUMERICAL DIFFERENTIATION AND INTEGRATION: Numerical Differentiation and Integration, Newton's forward, backward to find up to 2nd derivative. Newton cote's formula, trapezoidal and Simpson's rule (1/3 and 3/8), Romberg's method, Weddle's rule.

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

NUMERICAL SOLUTIONS FOR ORDINARY DIFFERENTIAL EQUATIONS: Difference Equations, order and degree, complementary function and particular integral of difference equation, Numerical solution of ODE- RK 4th order method to solve differential equation, Predictor- corrector methods(Milne's and Adam's) .

NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS: Graphical representation, classification of pde, SFPF AND DFPF, Liebmen's iteration, Bender Schmidt method, Crank – Nicholson method

PRACTICES:

- Estimate the Solution y at x=28from the following data using Stirling's formula.

x	20	25	30	35	40
f(x)	49225	48316	47236	45926	44306

- Using Lagrange's interpolation formula find y(10) from the following table:

x	5	6	9	11
f(x)	12	23	14	16

- Compute integral 0 to 4 f(x)dx using the Romberg integral technique on the trapezoidal integrals evaluated by the trapezoidal rule taking h = 1 and h = 0.5. The tabulated values are given below.

x	0	0.5	1	1.5	2.0	2.5	3.0	3.5	4.0
f(x)	1	4	3	2	2.5	2.9	3.6	4	1.8

- Use the Crank-Nicolson method and the central difference for the boundary conditions to solve the boundary value problem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, 0 < x < 1.\ u(x,0) = 2, 0 \le x \le 1, u(0,t) = 2, t \ge 0, \frac{\partial u}{\partial t}(1,t) = -u(1,t), t \ge 0$$

- Determine the solution of the partial differential equation by using Bender Schmidt method $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ subject to the conditions u(0,t) = u(5,t) = 0 and u(x,0) = x2(25-x2) taking h = 1 and k = 1/2, tabulate the values of u upto t=4 sec.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems.	Analyze	1	1,2,3
2	Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.	Apply	1	1,2,5
3	Apply numerical methods to obtain approximate solutions to mathematical problems.	Apply	2	1,2,3,5
4	Analyze and evaluate the accuracy of common numerical methods.	Analyze	2	1,2,3,5

TEXT BOOKS:

1. 1. B.S. Grewal, “Numerical Methods in Engineering & Science”, Khanna Publication, 9th edition.
2. P. Kandasamy, ‘’ Numerical methods”, S chand and company ltd India, Ist edition.

REFERENCE BOOKS:

1. E. Balagurusamy, “Numerical Method”, Tata McGraw Hill Publication.
2. S.S. Sastry, “Introductory Methods of Numerical Analysis”, PHI learning Pvt. Ltd.

II

YEAR

COMPUTER SCIENCE AND ENGINEERING

B.Tech.

COURSE CONTENTS

I SEM & II SEM

I SEMESTER

22ST202	-	Probability and Statistics
22TP201	-	Data Structures
22MS201	-	Management Science
22CS201	-	Database Management Systems
22CS202	-	Digital Logic Design
22CS203	-	Object Oriented Programming Through Java
22CT201	-	Environmental Studies
22SA201	-	Life Skills - I
	-	NCC/NSS/SAC/E-Cell/Student Mentoring/Social activities/Publication

II SEMESTER

22TP203	-	Advanced Coding Competency
22TP204	-	Professional Communication
22CS205	-	Computer Organization and Architecture
22CS206	-	Design And Analysis of Algorithms
22CS207	-	Operating Systems
22CS208	-	Theory of Computation
22SA202	-	Life Skills - II

22ST202 PROBABILITY AND STATISTICS

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

Hours Per Week :

L	T	P	C
3	0	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+0T+8P=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+0T+8P=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.

PRACTICES:

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

MODULE-2

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

REGRESSION ANALYSIS AND DISTRIBUTIIONS

Correlation and Regression: Correlation, types, Pearson's coefficient of correlation, regression, regression lines.

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

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

TESTING OF HYPOTHESIS

Testing large samples-one mean, two means, one proportion and two proportions. Testing small

SKILLS:

- ✓ *Collect the data from various data sources and evaluate mean, median, mode mean deviation and standard deviation.*
- ✓ *Identify the areas which we can apply the probability theory.*

samples- one mean, two means (independent and paired samples), Chi square tests-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	Blooms Level	Module No.	Mapping with POs
1	Apply measures of central tendency, skewness, and Karl Pearson's coefficient of skewness to study the statistical data sets.	Apply	1	1,2
2	Apply the probability theory and their applications to measure the uncertainty.	Apply	1	1,2
3	Study the relations between statistical variables and can fit the mathematical models for association.	Analyze	2	1,2,3
4	Test the statistical significances for various samples.	Evaluate	2	1,2,4
5	Identify the distribution type to measure the occurrences of chance.	Evaluate	2	1,4,5

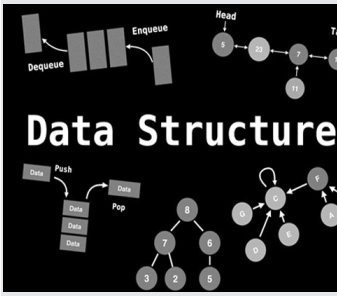
TEXT BOOKS:

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.

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 AND 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 **6L+6T+6P=18 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 **10L+10T+10P=30 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 successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Explore the organization of several ADTs and the manipulation (searching, insertion, deletion, traversing) of data stored in various data structures.	Apply	1,2	1
2	Apply different data structures to solve a given problem.	Apply	1,2	1
3	Analyze the efficiency of using different data structures and choose the efficient data structure for solving a given problem.	Analyze	1,2	2
4	Develop new algorithms to solve various problems.	Create	1,2	3,4

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.

Source:
https://xueqi326.wordpress.com/semester-3/management-science/

22MS201 MANAGEMENT SCIENCE

Hours Per Week :

L	T	P	C
2	2	0	3

PREREQUISITE KNOWLEDGE: Basic knowledge on management .

COURSE DESCRIPTION AND OBJECTIVES:

The goal of this course is to analyse 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.

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), Acceptance Sampling.

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.

SKILLS:

- ✓ *Expert in managerial skills.*
- ✓ *Maintain social relations.*
- ✓ *Evaluate pricing strategies.*

COURSE OUTCOMES:

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

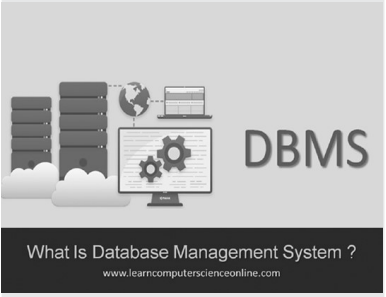
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze the nature and importance of management	Analyze	1	1,2,4,6
2	Significance of Operations Management.	Analyze	1, 2	1,2,5
3	Carry out production operations through work-study	Apply	1, 2	1, 2, 3, 5
4	Analyze the markets, customers, and competition	Analyze	2	1,2,4,5,6
5	Plan and control the HR function effectively	Evaluate	1, 2	1,2,3,4,5,6

TEXT BOOKS:

1. Stoner, Freeman, Gilbert, Management, 6th Ed, Pearson Education, New Delhi, 2004.
2. P. Vijay Kumar, N. Appa Rao and Ashnab, Chnalill, Cengage Learning India, 2012.

REFERENCES :

1. Kotler Philip & Keller Kevin Lane: Marketing Mangement 12/e, PHI, 2005.
2. Koontz & Weihrich: Essentials of Management, 6/e, TMH, 2005.
3. Thomas N. Duening & John M .Ivancevich Management - Principles and Guidelines, Biztantra, 2003.
4. Aryasri: Management Science, TMH, 2004.



Source: https://www.youtube.com/watch?v=lDpB9zF8LBw

22CS201 DATABASE MANAGEMENT SYSTEMS

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Discrete Mathematical 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, query optimization and transaction processing.

MODULE-1

UNIT–1 **8L+8T+8P=24 Hours**

DATABASE SYSTEM CONCEPTS

Databases And Database Users: Introduction; Characteristics of the database approach; Actors on the scene; Advantages of using 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.

UNIT–2 **8L+8T+8P=24 Hours**

RELATIONAL DATABASE DESIGN

Relational Database Design by ER–To-Relational Mapping: Relational Database design using ER-to-Relational mapping.

The Relational Data Model and Relational Database Constraints: Relational model concepts; Relational model constraints and Relational database schemas.

Relational Algebra: Unary relational operations - SELECT and PROJECT; Relational algebra operations from set theory; Binary relational operations - JOIN and DIVISION.

SQL: SQL data definition and data types; specifying constraints in SQL, Basic retrieval queries in SQL; INSERT, DELETE, and UPDATE statements in SQL.

PRACTICES:

- Design ER Model for various real time database applications.
- Development of Relational Database schemas for Company/Student/Sailors/ using DDL constructs of SQL.
- Apply various DML Commands such as select, insert, update etc. of SQL on Relational Database.
- Design of Relational Database schemas by specifying different types of Constraints.
- Apply various Relational Database operators (Arithmetic, Logical &comparison) and string-matching constructs of SQL.
- Expressing queries using Aggregate Functions of SQL on Relational Database.
- Queries on Relational Database using GROUP BY, HAVING and ORDER BY clauses of SQL.

MODULE-2

UNIT–1 **8L+8T+8P=24 Hours**

NORMALIZATION

Complex Queries, Triggers, Views: More complex SQL retrieval queries; Specifying constraints as assertions and actions as triggers; Views (virtual tables) in Pl/SQL.

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; Properties of relational decompositions, multi valued dependency, join dependencies.

UNIT–2 **8L+8T+8P=24 Hours**

TRANSACTION PROCESSING

Introduction To Transaction Processing Concepts and Theory: Introduction to transaction processing; Transaction and system concepts; Desirable properties of transactions; Characterizing schedules based on serializability.

Concurrency Control Techniques: Two-phase locking techniques for concurrency control, concurrency control based on timestamp ordering.

Database Recovery Techniques: Recovery concepts; Shadow paging; The ARIES recovery algorithm.

Indexing Structures for Files and Physical Database Design: Single level and multi-Level indexing; Dynamic multi-level indexing using B-trees and B+ trees.

PRACTICES:

- Design and Development of company database and expressing Nested queries using SQL.
- Design and Development of student database and specifying queries using set operations.
- Design and Development of sailor's database and specifying queries using different types of JOINs.
- Implementation of PL/SQL programs with Control Structures.
- Implementation of PL/SQL programs with Procedures.
- Implementation of PL/SQL programs with Function.
- Implementation of PL/SQL programs with Triggers.
- Creation and dropping of VIEWS.
- Relation R has eight attributes ABCDEFGH. Fields of R contain only atomic values. F = {CH -> G, A -> BC, B -> CFH, E -> A, F -> EG} is a set of functional dependencies (FDs) so that F+ is exactly the set of FDs that hold for R. How many candidate keys does the relation R have?
- Apply various DCL and TCL constructs of SQL on Relational Database.

SKILLS:

- ✓ *Develop E-R model for real life applications.*
- ✓ *Design of relational databases for real world applications.*
- ✓ *Devise queries using relational algebra and SQL.*
- ✓ *Analyze transaction processing, concurrency control and recovery techniques*

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Develop an E-R model for real life applications.	Apply	1	1,10
2	Design and normalize databases for real time applications.	Create	1	1,3
3	Devise queries using Relational Algebra and SQL.	Analyze	2	2
4	Express queries using database tools like Oracle, DB2, MYSQL.	Apply	2	5,10

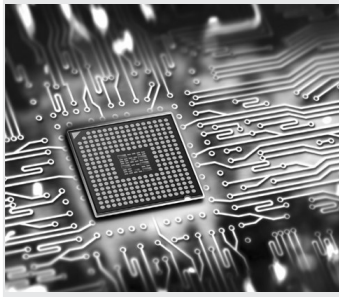
TEXT BOOKS:

1. Ramez, Elmasri and Shamkant B. Navathe, “Fundamentals of Database Systems”, 7th Edition, Pearson Education, 2016.
2. Raghu Rama Krishnan and Johannes Gehrke, “Database Management Systems”, 3rd Edition, Tata McGraw Hill, 2013.

REFERENCE BOOKS:

1. Abraham Silberschatz, Henry F.Korth and S.Sudarshan, “Database System Concepts”, 7th edition, Tata Mc Graw Hill,2019.
2. Allen G. Taylor “Database Development for Dummies” 1st Edition, 2011
3. C. J. Date “Introduction to Database Systems” 7th Edition, Addison Wesley, 2003.

22CS202 DIGITAL LOGIC DESIGN



Source: https://collegeacademy.in/images/ex-blogs/deld.jpg

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Basics of Computers.

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the basic knowledge on number systems, analysis and design of combinational and sequential circuits. The course mainly focuses on designing digital circuits in optimized manner by using components like decoders, encodes, multiplexers. It also deals with design of sequential circuits and Programmable logic devices.

MODULE-1

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

INTRODUCTION

Number Systems: Binary Numbers, Number base Conversions, Complements, Binary codes.

Boolean Algebra: Fundamental concepts of Boolean algebra basic theorems and properties.

Gate-Level Minimization: Canonical and standard forms - SOP and POS forms, Logic gates, Algebraic simplification and realization with basic gates and universal gates, The map method – two, three, four variable K map; POS and SOP simplification; Don't care conditions; NAND and NOR implementation.

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

COMBINATIONAL LOGIC CIRCUITS

Combinational circuits analysis, design procedure; Half adder, Full adder, Half subtractor, Full subtractor, Binary adder/subtractor; BCD adder; Binary multiplier; Magnitude comparator; Decoders; Encoders; Multiplexers; De-Multiplexer.

PRACTICES:

- Design a combinational circuit with three inputs and one output. The output is 1 when the binary value of the inputs is less than 3. The output is 0 otherwise.
- Design a combinational circuit with three inputs x, y, z and three outputs A, B, C. When the binary inputs is 0, 1, 2 or 3, the binary output is one grater than the input. When the binary input is 4, 5, 6, or 7 then the binary output is one less than the input.
- Design a code converter that converts a decimal digit from the 8, 4, -2, -1 code to BCD.
- Implement a Full – Adder using 4 X 1 multiplexer.
- Design a 16 X 1 Multiplexer with five 4 X 1 multiplexer.
- Design a 5-to-32 line decoder with four 3-to-8-line decoders with enable and 2-to-4-line decoder.
- Design and realize 4bit parallel adder and subtractor.

MODULE-2

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

CIRCUITS, REGISTERS AND COUNTERS

Sequential Logic Circuits: Latches, Flip-Flops - SR, JK, D, T; Flip-flop conversion; Analysis of sequential circuits; Design procedure.

Registers and Counters: Shift registers; Ripple counters; Synchronous counters.

SKILLS:

- ✓ *Learn different data and number representations.*
- ✓ *Design of logical circuits using all types of gates.*
- ✓ *Minimizing of Boolean functions.*
- ✓ *Design of simple logical circuits*

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

MEMORY AND PROGRAMMABLE LOGIC DEVICES

Random access memory; Read only memory; Programmable logic array; Programmable array logic.

PRACTICES:

- Design a JK flip-flop using a D flip-flop.
- Design a sequential circuit with two D flip-flops A and B and, one input x. When x=0, the state of the circuit remains same. When x = 1, the circuit goes through the state transitions from 00 to 01 to 11 to 10 back to 00 and repeats.
- Design a 4-bit binary synchronous counter with D flip-flop.
- Design and realize a decimal counter.
- A sequential circuit has two J-K fiip-flops A and B. Two inputs x and y, and one output z. The flip-flop input equations and circuit output equation are:
  JA=Bx+B'y'
  KA=B'xy'
  JB=A'x
  KB=A+xy'
  Z=Ax'y'+Bx'y'
  a) Tabulate the state table. b) Derive the state equations.
- Realize the given two Boolean functions with a PLA:
  F1(A, B, C) = ∑(0, 1, 2, 4)
  F2(A, B, C) = ∑ (0, 5, 6, 7)
  F3(A, B, C) = ∑ (1, 3, 4, 5, 7)
- Tabulate the PAL programming table for the four Boolean functions listed below. Minimize the numbers of product terms.
  A(x, y, z) = ∑ (1, 3, 5, 6) B(x, y, z) = ∑ (0, 1, 6, 7)
  C(x, y, z) = ∑ (3, 5) D(x, y, z) = ∑ (1, 2, 4, 5, 7).

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply the knowledge of digital logic concepts to optimize digital circuits.	Apply	1	1
2	Apply Boolean algebra rules and karnaugh map method to reduce the Boolean functions.	Apply	1	1, 2
3	Design Combinational digital circuits for the given problem statement by applying the digital techniques.	Analyze	1	3
4	Design and analyze sequential digital circuits for the given problem statement and improve the performance by reducing the complexities.	Analyze	2	3
5	Categorize various types of Programmable Logic Devices.	Analyze	2	2

TEXT BOOK:

1. M Morris Mano and Michael D. Ciletti, "Digital Design", 5th Edition, Pearson Education, 2013.

REFERENCE BOOKS:

1. John F.Wakerly, "Digital Design Principles and Practices", 3rd Edition, Pearson/PHI, 2015
2. Charles H.Roth. "Fundamentals of Logic Design", 6th Edition, Thomson Learning, 2013.
3. John. M Yarbrough, "Digital Logic Applications and Design", Thomson Learning, 2006.

22CS203 OBJECT-ORIENTED PROGRAMMING THROUGH JAVA

Source: https://www.datasciencecentral.com/wp-content/uploads/2021/10/8667507462.jpeg

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Classes, Objects and Inheritance.

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 **8L+0T+8P=16 Hours**

INTRODUCTION

History of Java, Byte code, JVM, Java buzzwords, OOP principles, Data types, Variables, Scope of variables, Operators, Control statements, Type conversion and casting, Arrays.

Concepts Of Classes and Objects: Introduction to methods, Method over loading, Constructors, Constructor overloading, Usage of static with data and method, Access control, this keyword, Garbage collection, String class, String Tokenizer.

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

INHERITANCE AND EXCEPTIONS

Inheritance: Types of inheritance, Member access rules, Usage of super keyword, Method overriding, Usage of final keyword, Abstract classes, Interfaces - differences between abstract classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Packages-defining, creating and accessing a package, importing packages, access control in packages.

Exception Handling: Concepts of exception handling, Types of exceptions, Usage of try, catch, throw, throws and finally keywords, Built-in exceptions, User defined exception.

PRACTICES:

- There is a telecommunication company called “Powered Air” who have approached you to build their Interactive Voice Response (IVR) system. write a Java program and be able to provide the following menu (given below):

  Note: User should provide an input for each menu display. Welcome to Powered Air service. What would you like to do?

  a. Know my balance. b. Know my validity date

  c. Know number of free calls available. d. More

  1. Prepaid Bill Request 2. Customer Preferences 3. GPRS activation

  4. Special Message Offers 5. Special GPRS Offers 6. 3G Activation

  7. Go back to Previous menu

  You are free to display your own messages in this IVR.
- Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.

SKILLS:

- ✓ *To analyse and develop algorithm for real life problems using Java.*
- ✓ *Experience with developing and debugging programs in different IDEs.*
- ✓ *Develop multi-threaded applications.*
- ✓ *Creating web applications*

Hint: Area of rectangle = length * width, Perimeter of rectangle = 2*(length+width).

- Implement a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (use StringTokenizer class).
- Implement a java program to print all tokens of a string on the bases of multiple separators (use StringTokenizer class).
- Using inheritance, one class can acquire the properties of others. Consider a class Animal that has only one method “walk”. Next, create a Bird class that also has a fly method. Finally, create a bird object that can both fly and walk.
- Using inheritance, Write the following code in your editor :

  1. A class named Arithmetic with a method named “add” that takes integers as parameters and returns an integer denoting their sum.

  2. A class named Adder that inherits from a superclass named Arithmetic.

  Note: Your classes should not be Public.
- When a subclass inherits from a superclass, it also inherits its methods; however, it can also override the superclass methods (as well as declare and implement new ones). Consider the Sports class having methods getName()[which returns name of sport] and getNumberOf TeamMembers()[which returns noof team members] create a Soccer class that inherits from the Sports class. We can override the get Name method and return a different subclass-specific string and override getNumberOfTeamMembers method and return noof team members
- Implement a java program to create an abstract class named Shape that contains an empty method named number Of Sides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method number Of Sides ( ) that shows the number of sides in the given geometrical figures.
- You are given an interface Advanced Arithmetic which contains a method signature int divisor_ sum(int n). You need to write a class called My Calculator which implements the interface. divisor_sum function just takes an integer as input and return the sum of all its divisors. For example divisors of 6 are 1, 2, 3 and 6, so divisor_sum should return 12. The value of n will be at most 1000.
- Implement a Java program for the following

  a) Creation of simple package.

  b) Accessing a package.
- Implement a Java program to read two numbers a,b from user and perform division a/b,if the user passes b value as zero, handle the exception using try and catch otherwise display the result.
- Create a class called Customer with data members account_number, balance (initialize with 10000), and member functions print(), deposit(), and withdraw(). Print method display account number and balance. If withdraw amount is less than current balance while withdrawing, throw an exception “In Sufficient Funds”. If the input is 1 do print. If the input is 2 withdraw (). If the input is 3 deposit. If the input is 4 terminate program.
- Implement a Java program which accepts age as input from the user and throws an exception “Not Eligible to Vote” when age is <=18 otherwise print “Eligible to Vote”.

MODULE-2

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

MULTI THREDING AND FRAMEWORK

Multithreading: Concepts of multi threading, Differences between process and thread, Thread life cycle, Creating multiple threads using Thread class and Runnable interface, Synchronization, Thread priorities, Inter thread communication.

Collection Framework: Collections Overview, Collection Interfaces - List, Set, Map, List – Array List, Linked List, Vector, Set - HashSet, TreeSet, Map - HashTable, HashMap, accessing a collection via an Iterator, comparator, comparable.

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

SWINGS

GUI Programming With Swing: Delegation event model-Events, Event sources, Event Listeners, Event classes, handling mouse and keyboard events.

Exploring Swing Controls: JLabel and Image Icon, JText Field, JButton, JCheckBox, JRadioButton, JTabbed Pane, JList, JCombo Box.

PRACTICES:

- Print in Order

  Suppose we have a class:

  public class Foo {

  public void first() { print("first"); }

  public void second() { print("second"); }

  public void third() { print("third"); }

  }

  The same instance of Foo will be passed to three different threads. Thread A will call first(), thread B will call second(), and thread C will call third(). Design a mechanism and modify the program to ensure that second() is executed after first(), and third() is executed after second().

  Note: We do not know how the threads will be scheduled in the operating system, even though the numbers in the input seem to imply the ordering. The input format you see is mainly to ensure our tests' comprehensiveness.

  Example 1:

  Input: nums = [1,2,3] Output: "firstsecondthird"

  Explanation: There are three threads being fired asynchronously. The input [1,2,3] means thread A calls first(), thread B calls second(), and thread C calls third(). "firstsecondthird" is the correct output.

  Example 2:

  Input: nums = [1,3,2] Output: "firstsecondthird"

  Explanation: The input [1,3,2] means thread A calls first(), thread B calls third(), and thread C calls second(). "firstsecondthird" is the correct output.

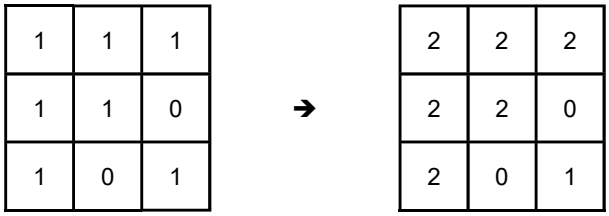
- **Flood Fill:**

  An image is represented by an m x n integer grid image where image[i][j] represents the pixel value of the image.

  You are also given three integers sr, sc, and color. You should perform a flood fill on the image starting from the pixel image[sr][sc].

  To perform a flood fill, consider the starting pixel, plus any pixels connected 4-directionally to the starting pixel of the same color as the starting pixel, plus any pixels connected 4-directionally to those pixels (also with the same color), and so on. Replace the color of all of the aforementioned pixels with color.

  Return the modified image after performing the flood fill.



  Example 1:

  Input: image = [[1,1,1],[1,1,0],[1,0,1]], sr = 1, sc = 1, color = 2

  Output: [[2,2,2],[2,2,0],[2,0,1]]

  Explanation: From the centre of the image with position (sr, sc) = (1, 1) (i.e., the red pixel), all pixels connected by a path of the same color as the starting pixel (i.e., the blue pixels) are colored with the new color.

Note the bottom corner is not coloured 2, because it is not 4-directionally connected to the starting pixel.

Example 2:

Input: image = [[0,0,0],[0,0,0]], sr = 0, sc = 0, color = 0

Output: [[0,0,0],[0,0,0]]

Explanation: The starting pixel is already colored 0, so no changes are made to the image.

- Count words in a given string

  The input parameter is a list of strings representing lines of text.

  Count how often the word occurs in the text.

  If the word "kitten" occurred in a text 23 times, then its entry would be "kitten - 23\n". Return statistics as a String containing all the entries.

  Omit all words which contain less than 4 letters and appear less than 10 (the words which are too small or to rare) The entries in the resulting String should be also sorted by their amount and then in alphabetical order if it is needed.
- Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
- Implement a Java program for handling key events when the key board is pressed, released, typed.
- Implement a Java swing program that reads two numbers from two separate text fields and display sum of two numbers in third text field when button "add" is pressed.
- Implement a Java program to design student registration form using Swing Controls. The form which having the following fields and button "save". Form Fields are: Name, RNO, Mail id, Gender, Branch, and Address.
- Implement a java program using swings to design a multiple choice question having three options (use radio button) ,display the message using dialog box "Your answer is wrong" if the user selects wrong option otherwise display ,"Your answer is correct."

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply object oriented concepts on real time scenarios.	Apply	1	1,2
2	Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes	Apply	1, 2	1,2
3	Design and develop Java applications to solve real world problems by using modern tools and collection framework	Create	2	3,5
4	Design and develop GUI based applications using swings for internet and system based applications.	Create	2	3,5

TEXT BOOKS:

1. Herbert Schildt, "Java the complete reference", 12th Edition, McGraw Hill, Education, 2021.
2. M.T. Somashekara, D. S. Guru, K.S. Manjunatha, "Object-Oriented Programming with Java", 1st Edition, PHI Learning, 2017.

REFERENCE BOOKS:

1. E. Balagurusamy, "Programming with Java", 6th Edition, McGraw Hill, 2019.
2. Mark Lassoff," Java Programming for Beginners: Learn the fundamentals of programming with Java", 1st Edition, Packt Publishing Limited,2017.
3. Philip Conrod, Lou Tylee," Learn Java GUI Applications : A JFC Swing Tutorial", 11th Edition, Kidware Software, 2019.

22CT201 ENVIRONMENTAL STUDIES

Source: Biogas plant at VFSTR

Hours Per Week :

L	T	P	C
2	1	0	2

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

COURSE DESCRIPTION AND OBJECTIVES:

It is a multidisciplinary subject where different aspects of society and environment are dealt 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 for planning appropriate strategies to address environmental issues. It identifies and creates solutions that conserve to manage ecosystem and biodiversity and helps to eliminate pollutants, toxicants, 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 **8L+4T+0P=12 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 **8L+4T+0P=12 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: river / pond / lake / forest / grassland / hill / mountain and study of different types of ecosystems, biodiversity study and documentation (herbarium sheet preparation).
- Set up an aquarium.
- Case study: Renewable energy use.

MODULE-2

UNIT-1 **8L+4T+0P=12 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 **8L+4T+0P=12 Hours**

POLLUTION CONTROL DEVICES AND WASTEWATER 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 wastewater 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	Blooms Level	Module No.	Mapping with POs
1	Apply the basic concepts of sustainable development, natural resource utilization and ecology for the purpose of environmental protection.	Apply	1	1,6,7, 9, 10, 11, 12
2	Design remediation technologies for their abatement	Apply	2	1, 3,6,7, 9, 10, 11, 12
3	Analyze the biodiversity of different ecosystems and formulate various conservation approaches	Analyze	1	1, 7, 8, 9, 10, 11, 12
4	Analyze the presence of various environmental pollutants.	Analyze	2	1, 6,7,9, 10, 11, 12
5	Recommend various waste management approaches and their implementation strategies.	Evaluate	2	1,2, 7,8,9,10,11, 12

TEXT BOOKS:

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.

22TP203 ADVANCED CODING COMPETENCY

Hours Per Week :

L	T	P	C
0	0	2	1

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

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.

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**

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.

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:

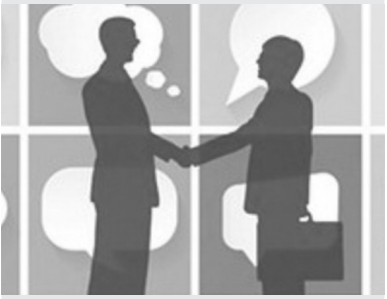
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply various data structures to solve a different algorithm.	Apply	1,2	1
2	Investigate the various data structures to solve a given problem in an efficient manner.	Analyse	1,2	2
3	Design and implement an appropriate hashing function for an application.	Create	1,2	4

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.



Source: https://www.coursera.org/specializations/improve-english

22TP204 PROFESSIONAL COMMUNICATION

Hours Per Week :

L	T	P	C
0	0	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.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Possess comprehensive skills in listening and reading business texts in formal context.	Apply	2	7
2	Communicate effectively both in their academic as well as professional environment.	Apply	1, 2	10
3	Clear grasp on the register of business language.	Analyze	1	8
4	Possess the ability to write business reports and proposals clearly and precisely to succeed in their future.	Create	1	12
5	Make effective presentations and participate in formal context.	Create	2	10

TEXT BOOK:

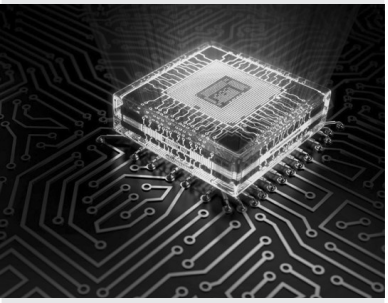
1. S. Schnurr, "Exploring Professional Communication: Language in Action", London: Routledge, 2013.

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.

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.*
- ✓ *Know and practice specific functions and vocabulary in a business context.*
- ✓ *Produce short business reports, proposals and correspondence.*
- ✓ *Write various business documents through reading techniques.*



Source: https://machinelearningmedium.com/assets/images/computer-architecture.jpg

22CS205 COMPUTER ORGANIZATION AND ARCHITECTURE

Hours Per Week :

L	T	P	C
2	2	0	3

PREREQUISITE KNOWLEDGE: Digital logic design.

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.

MODULE-1

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

INTRODUCTION, RTL, DATA REPRESENTATION AND COMPUTER ARITHMETIC

Introduction, Register Transfer language & Data Representation: Organization and Architecture, Register Transfer, Bus and Memory Transfers, Data Representation-Fixed Point Representation, Floating Point Representation.

Computer Arithmetic: Fixed point arithmetic operations such as Addition and Subtraction, Multiplication Algorithms, Division Algorithms.

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

MICRO OPERATIONS AND BASIC COMPUTER ORGANIZATION AND DESIGN

Micro operations: Arithmetic Micro operations, Logic Micro Operations, Shift Micro Operations, Arithmetic Logic Shift Unit.

Basic Computer Organization and Design: Instruction Codes, Computer Register, Computer Instructions, Instruction Cycle, Memory – Reference Instructions. Register Reference Instructions, Input –Output and Interrupt.

PRACTICES:

- Design a Common bus system for eight registers with eight bits each using multiplexers.
- Design a Common bus system for four registers with four bits each using three state gate buffers.
- A digital computer has a common bus system for 16 registers of 32 bits each. The bus is constructed with multiplexers.
  How many selection inputs are there in each multiplexer?
  What size of the multiplexers are needed?
  How many multiplexers are there in the bus?
- Perform arithmetic operations (+42) + (-13) and (-42) – (-13) in binary using signed 2's complement representation for negative numbers.
- Find the product using Booth Multiplication Algorithm.
- a. (9) X (13) b. (9) X (-13) c. (-9) X (13) d. (-9) X (-13)
- Perform the division of 27 and 4 using Division algorithm.
- Design a 4- bit combinational circuit decrementer using 4 full adder circuits.
- Register A holds the 8-bit binary 11011001. Determine the B operand and the logic micro operation to be performed in order to change the value in A to :
  a) 01101101 b) 11111101

- An 8-bit register contains the binary value 10011100. What is the register value after an arithmetic shift right? Starting from the initial number 10011 100, determine the register value after an arithmetic shift left, and state whether there is an overflow.
- Starting from an initial value of R =11011101, determine the sequence of binary values in R after a logical shift-left, followed by a circular shift-right, followed by a logical shift-right and a circular shift-left.
- Design arithmetic logic shift unit that performs different operations on 4 bits.

SKILLS:

- ✓ *Learn different data representations.*
- ✓ *Design digital circuitry for implementing different operations.*
- ✓ *Identify the types of memories and their uses.*
- ✓ *Study various data transfer mechanisms in digital computer and I/O.*

MODULE-2

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

CPU AND MEMORY ORGANIZATION

Central Processing Unit: General Register Organization, STACK Organization. Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory.

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

I/O ORGANIZATION

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupts, Direct Memory Access.

PRACTICES:

- The content of PC in the basic computer is 3AF (all numbers are in hexadecimal). The content of AC is 7EC3. The content of memory at address 3AF is 932E. The content of memory at address 32E is 09AC. The content of memory at address 9AC is 8B9F.

  What is the instruction that will be fetched and executed next?

  Show the binary operation that will be performed in the AC when the instruction is executed.

  Give the contents of registers PC, AR, DR, AC, and IR in hexadecimal and the values of E, I, and the sequence counter SC in binary at the end of the instruction cycle.
- Implement the given expressions into different addressing architectures.

  Y=(A-B)/(C*D + E) b. Y=A-B+C*(D *E+F)
- How many 128 x 8RAM chips are needed to provide a memory capacity of 2048 byte?
- How many lines of the address bus must be used to address 2048 bytes of memory? How many of these lines will be common to all chips?
- How many lines must be decoded for chip select and design the size of the decoders.
- A computer uses RAM chips or 1024 x 1 capacity.

  How many chips are needed, and show the connection of memory capacity 1024 bytes?

  How many chips are needed to provide a memory capacity or 16K bytes? Explain in words how the chips are to be connected to the address bus.
- How many characters per second can be transmitted over a 1200-baud line in each of the following modes? (Assume a character code of eight bits.)

  Synchronous serial transmission.

  Asynchronous serial transmission with two stop bits.

  Asynchronous serial transmission with one stop bit.
- Information is inserted into a FIFO buffer at a rate of m bytes per second. The information is deleted at a rate of n byte per second. The maximum capacity of the buffer is k bytes.

  How long does it take for an empty buffer to fill up when m >n?

  How long does it take for a full buffer to empty when m <n?

  Is the FIFO buffer needed if m = n?

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze Computer Organization and Computer Architecture, different arithmetic operations.	Analyze	1	1, 2, 12
2	Design different digital circuits required to perform the micro operations.	Apply	1	1, 2, 3, 12
3	Design interface circuits for memory and peripheral, DMA and communication devices. Compare various modes of transfer.	Analyze	2	1, 2, 3, 4, 12
4	Evaluate the performance of a processor and memory in terms of speed, size and cost.	Evaluate	2	1, 2, 12

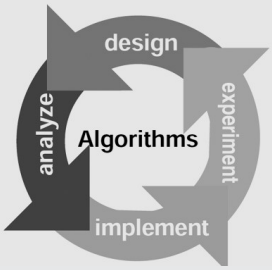
TEXT BOOKS:

1. M. Morris Mano, “Computer System Architecture”, 3rd Edition update, Pearson, 2017.
2. William Stallings, “Computer Organization & Architecture: Designing for Performance”, 11th Edition, Pearson, 2019.

REFERENCE BOOKS:

1. Carl Hamacher, ZvonkoVranesic, SafwatZaky, “Computer Organization”, 5th Edition, McGraw Hill, 2002.
2. Vincent P. Heuring and Harry F Jordan, “ Computer Systems Design and Architecture “, 2nd edition , Pearson/ Prentice Hall India 2004.
3. David A. Patterson and John L. Hennessy, “Computer Organization and Design-The Hardware/ Software Interface”, ARM Edition, 5th Edition, Elsevier, 2009.

22CS206 DESIGN AND ANALYSIS OF ALGORITHMS



Source: https://www.facebook.com/Design-and-Analysis-of-Algorithms-1553902878155564/

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Programming for problem solving, Discrete Mathematical Structures, Data Structures.

COURSE DESCRIPTION AND OBJECTIVES:

This course offers the basic knowledge required to analyze the asymptotic performance of algorithms. In addition, this course provides the knowledge required to solve different problems using suitable design strategies such as the greedy method, divide and conquer, dynamic programming, backtracking and branch & bound. This course helps to understand the impact of the choice of data structures and algorithm design strategies on the performance. This course also provides the understanding of advanced graph applications and throws light on tractable and intractable problems.

MODULE-1

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

INTRODUCTION

Algorithm, Pseudo-code for expressing algorithms, Performance analysis – space and time complexity; Asymptotic notation - big oh notation, Omega notation, Theta notation and little oh notation; Analysis of recursive algorithms through recurrence relations- substitution method, Recursion tree method, Masters Theorem.

Disjoint sets: Disjoint set operations, Union and find algorithms.

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

DIVIDE & CONQUER AND GREEDY METHOD

Divide and Conquer: General method, Applications - Binary search, Quick sort, Merge sort and Strassen's matrix multiplication.

Greedy Method: Applications - job sequencing with deadlines, Knapsack problem, Minimum cost spanning trees.

PRACTICES:

- Sort a given set of elements using the following methods and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n inputs. The elements can be read from a file or can be generated using the random number generator.
  a. Quick sort b. Merge sort
- Search for a given set of elements using the following methods and determine the time required to search the given element. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus no. of elements. The elements can be read from a file or can be generated using the random number generator.
  a. Linear Search b. Binary Search
- Implement the following using divide and conquer approach.
  To multiply two given square matrices.
  To multiply two given square matrices using Strassen's matrix multiplication.
- Design the Algorithm to solve Job sequencing with deadlines problem and Analyze its time complexity. Implement the above algorithm using Greedy method.

SKILLS:

- ✓ *Analyze the given algorithm concerning space and time complexities and compare it with other algorithms.*
- ✓ *Develop algorithms for solving problems using divide and conquer, greedy, dynamic programming, backtracking and branch & bound techniques.*
- ✓ *Application of existing design strategies to solve real-world problems.*

- Design the Algorithm to solve fractional Knapsack problem using Greedy method. Analyze the time complexity and implement the above algorithm.
- Design the Algorithm to find minimum spanning tree and its cost for an undirected graph. Analyze the time complexity and implement the above algorithm.

MODULE-2

UNIT-1 **10L+10T+10P=30 Hours**

DYNAMIC PROGRAMMING AND BACKTRACKING

Dynamic Programming: General method, Applications - optimal binary search trees, Matrix chain multiplication, 0/1knapsackproblem, All pairs shortest path problem, Travelling sales person problem.

Backtracking: General method, Applications - N-Queen problem, Sum of subsets problem, Graph colouring and Hamiltonian cycles.

UNIT-2 **6L+6T+6P=18 Hours**

BRANCH & BOUND AND P, NP, NP - HARD AND NP-COMPLETE

Branch and Bound: General method, Applications- Travelling sales person problem, 0/1 knapsack problem using LC branch and bound solution and FIFO branch and bound solution.

P, NP, NP - HARD and NP-Complete: Basic Concepts - Non-Deterministic Algorithms - The Classes NP-Hard and NP Complete- NP Hard Problems- Clique Decision Problem-Cook's Theorem.

PRACTICES:

- Design the Algorithm to find all pairs shortest path problem by using dynamic programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to find optimal binary search tree and its cost by using dynamic programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to find optimal order of matrix chain multiplication and its cost using dynamic programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to find optimal route for travelling sales person problem and its cost by using dynamic Programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to solve N-queens problem by using backtracking approach and Analyze its time complexity. Implement the above algorithm.
- Design the Algorithm to solve sum of subsets problem using backtracking approach and Analyze its time complexity. Implement the above algorithm.
- Design the Algorithm to solve 0/1 Knapsack problem using Branch and Bound method. Analyze the time complexity and Implement the above algorithm.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze the efficiency of a given algorithm using time and space complexity theory. Understanding algorithmic design strategy like divide and conquer approach.	Analyze	1	1, 2, 12
2	Apply greedy algorithm Strategy for suit able problems and argue the correctness of such algorithms with respect to the global optimization.	Apply	1	1, 2,3, 5, 12
3	Apply the dynamic programming paradigm and identify the kind of problem best suited to solve using dynamic programming.	Apply	2	1, 2, 3, 5, 12
4	Compare and contrast the design principles of branch and bound with backtracking strategy.	Apply	2	1, 2,3,5, 12
5	Investigate computational complexity of different class of problems.	Analyze	2	1, 2,4,12

TEXT BOOKS:

1. Ellis Horowitz, SatrajSahni and Rajasekharan, “Fundamentals of Computer Algorithms”, 2nd Edition, Galgotia publications, 2006.
2. Thomas H. Coremen, Charles E. Leiserson and Ronald L. Rivest, “Introduction to Algorithm”, 2nd Edition, MIT press Ltd., 2014.

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”, 2nd Edition, Addison Wesley Publishing Company, 1998.
3. Ronald L. Graham, Donald E. Knuth and Oren Patashnik, “Concrete Mathematics”, 2nd Edition, Addison wesley Publishing Company,1998.
4. Dasgupta, Papadimitriou and Vazirani,“Algorithms”, 1st Edition, McGraw-Hill publishers, 2008.
5. Weiss, “Data Structures and Algorithm Analysis”, 1st Edition, Addison-Wesley Publishing Company, 2016.

Source: https://www.123rf.com/stock-photo/operating_system.html

22CS207 OPERATING SYSTEMS

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Knowledge of computers fundamentals, Computer organization & Digital logic and its design.

COURSE DESCRIPTION AND OBJECTIVES:

This course aims at concepts and principles of Operating Systems, its overall responsibility inacting as an interface between the system's hardware components and the user. Further, it also helps students to understand the different scheduling policies, process synchronization mechanisms, deadlock handling mechanisms and memory management techniques.

MODULE-1

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

LINUX FILE SYSTEM & PROCESS SCHEDULING

Introduction to LINUX File System: The LINUX file System, File System Hierarchy, File system Commands, File Attributes, File Permissions.

Filters: cmp, comm, diff, head, tail, find, cut, paste, sort, uniq.

Regular Expressions: grep, egrep, fgrep, Sed- line addressing, context addressing, text editing, substitution.

Introduction to Operating System: What Operating System do; Operating System Structure; Process concept-overview, Process Scheduling, Operations on Process; Inter Process Communication; Threads;

Process (CPU) Scheduling-Scheduling Criteria, Scheduling Algorithms; Multiple-Processor scheduling;

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

PROCESS SYNCHRONIZATION AND DEADLOCKS

Process Synchronization: The critical-section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

Deadlocks: Deadlock characterization; Methods of handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery.

PRACTICES:

- Use the cat command to create a file containing the following data. Call it mytable.txt usetabsto separate the fields.

  1425 ravi 15.65
  4320 ramu 26.27
  6830 sita 36.15
  1450 raju 21.86

  a. Use the cat command to display the file, mytable.txt.
  b. Use the vicomm and to correct any errors in the file, mytable.txt.
  c. Use the sort command to sort the file mytable.txt according to the first field.
  d. Call thesortedfilemytable.txt (same name)
  e. Printthefilemytable.txt.
  f. Use the cut &paste commands to swap fields 2and 3mytable.Call itmytable.txt (same name)

g. Print the new file, mytable.txt.

- Write a shell script that takes a command–line argument and reports on whether it is directory, a file, or something else.
- Write a shell script that accepts one or more file name as arguments and convertsall of them to uppercase, provided they exist in the current directory.
- Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
- Write a shell script that computes the total and average marks of a student according to the following;
- Ifaveragemarks≥69thenresultis-Distinction‖.
- Ifaveragemarks≥59and≤70thenresultis-FirstClass‖.
- Ifaveragemarks≥49and≤60thenresultis-SecondClass‖If average marks ≤50 then result is -Pass‖.
- Note that any subject marks ≤ 40then result is-Fail‖.
- Accept student name and six subject marks through the keyboard.
- Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.
- Write a shell script, which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.
- Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.
- Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
- Implementation of new process creation and its communications.
- Implement of thread creation and deletion.
- Implementation of FCFS scheduling.
- Implementation of SJF and RR Scheduling.
- Implementation of producer consumer problem.
- Implementation of Banker's algorithm for Dead lock avoidance.

SKILLS:

- ✓ *Manage open-source operating systems like Ubuntu, Fedora etc.*
- ✓ *Know the concepts of Processes scheduling and File Systems.*
- ✓ *Identification of different disk scheduling methodologies.*

MODULE-2

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

MEMORY MANAGEMENT

Memory Management: Basic concept tofmemory management, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation.

Virtual Memory Management: Demand Paging, Page Replacement: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU), Least Recently used (LRU), Allocation of Frames.

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

SECONDARY STORAGE STRUCTURE

Secondary Storage Structure: Over view of mass-storage structure, disk structure, disk scheduling;

File System Interface - File concept, Access Methods, Directory & Disk Structure, File-System Mounting, File Sharing, Protection; File-system structure.

File System Implementation- Directory implementation, Allocation Methods, Free Space Management.

PRACTICES:

- Assume that you have a page-reference string for a process with m frames (initially all empty). The page-reference string has length p, and n distinct page numbers occur in it.
  a) What is a lower bound on the number of page faults?
  b) What is an upper bound on the number of page faults?
- Consider the following page-replacement algorithms. Rank these algorithms on a five-point scale from "bad" to "perfect" according to their page-fault rate. Separate those algorithms that

suffer from Belady's anomaly from those that do not.

a) LRU replacement. b) FIFO replacement.

c) Optimal replacement. d) Second-chance replacement.

- Consider the page reference string: 1, 2, 3, 4,2, 1, 5, 6, 2, 1, 2, 3,7, 6, 3, 2, 1, 2, 3, 6.
- How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.
- LRU replacement.
- FIFO replacement.
- Optimal replacement.
- How many page fault soccur for your algorithm for the following reference string with four page frames? 1, 2, 3,4, 5, 3, 4, 1, 6,7, 8, 7, 8, 9, 7, 8,9, 5, 4, 5,4, 2.
- What is the minimum number of page faults for an optimal page replacement strategy for thereference string above with four page frames?
- Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of the CPU and the paging disk. Three alternative results are shown below. For each case, what is happening?

  a) Can the degree of multiprogramming be increased to increase the CPU utilization? Is the paging helping?

  b) CPU utilization 13 percent; disk utilization 97 percent.

  c) CPU utilization 87percent; disk utilization 3 percent.

  d) CPU utilization 13 percent; disk utilization 3 percent.
- Implementation of Disk scheduling algorithm–FCFS.
- Implementation of Disk scheduling algorithm–SSTF and SCAN.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Classify the basic concepts of operating system and explore Linux ecosystem.	Analyze	1	1
2	Apply the concepts of process scheduling algorithms and process synchronization techniques to derive the efficiency of resource utilization.	Apply	1	1, 2, 3, 5, 12
3	Analyze the requirements for attempting Operating systems principles.	Analyze	1,2	1,2,12
4	Design the various memory management schemes For a given scenario.	Create	2	3,5
5	Apply the concepts of file system interface and implementation.	Apply	1,2	2,5

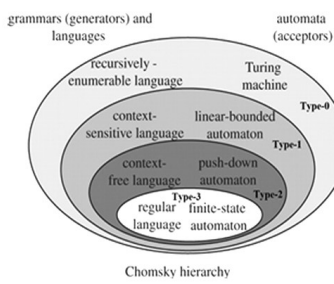
TEXT BOOKS:

1. Sumitabha Das, Unix concepts and applicationsǁ, TMH Publications, 4th Edition, July 2017.
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", 9th Edition, John Wiley & SonsInc, 2013.

REFERENCE BOOKS:

1. Richard. Stevens and Stephen A Rago, "Advanced Programming in the Unix Environment", 3rd Edition, Addison-Wesley, 2013.
2. William Stallings, "Operating Systems-Internals and Design principles" PHI, 7th Edition, 2012.
3. Gary J. Nutt. Addison-Wesley, "Operating Systems: A Modern Perspective", 2nd Edition, 2001.
4. B.A. Forouzan & R.F.Giberg, - Unix and shell Programmingǁ, Thomson, 1st Edition, New Delhi, 2003.

22CS208 THEORY OF COMPUTATION



Source: https://sameer9247.wordpress.com/2016/11/15/theory-of-computation/

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Knowledge of graphs, trees and logic.

COURSE DESCRIPTION AND OBJECTIVES:

This course aims to teach the student to identify different formal language classes and their relationships, strong theoretical foundation for designing compilers. In addition to this the student will be able to learn the techniques for information processing, design different grammars, automata and recognizers for different formal languages.

MODULE-1

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

INTRODUCTION

Alphabets, Strings and languages, Automata and Grammars, Regular languages, Chomsky hierarchy of 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, FA with output - Moore and Mealy machine, Equivalence of Moore and Mealy machine, Applications and Limitation of FA.

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

REGULAR EXPRESSIONS

Regular Expression (RE): Definition, Operators of regular expression and their precedence, Algebraic laws for Regular Expressions, Kleen's Theorem, Regular Expression to FA, DFA to regular expression, Arden theorem, Non regular languages, pumping lemma for regular languages (proofs not Required), Application of pumping lemma, Closure properties of regular languages, Decision properties of regular languages.

Grammar Formalism: Regular Grammars-Right linear and left linear grammars, Equivalence between regular linear grammar and FA;

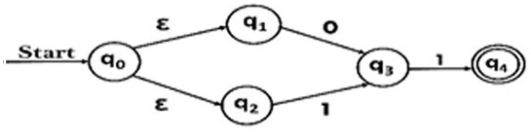
PRACTICES:

- Design DFA and NFA which accepts the following languages over the alphabet {0,1}. And also covert NFA to DFA. Give separate Automata for each and also write RE for the obtained automata.
  a) The set of all strings ends with 00.
  b) With three consecutive 0's.
  c) With 011 as a substring.
  d) Either begin or ends with 01.
  e) Strings whose fourth symbol from the right end is 1.
  f) Even number of 0's.
  g) number of 1 's are divisible by three.
- Design NFA to recognize the following set of strings.
  a) abc, abd, and aacd: Assume the alphabet is {a,b,c,d}.
  b) 0101,101 and 011: Assume the alphabet is {0,1}.
  c) ab,bc and ca: Assume the alphabet is {a,b,c}.

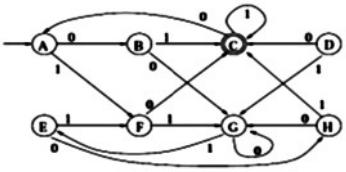
SKILLS:

- ✓ *Investigate syntax and semantics of a regular and context free languages.*
- ✓ *Develop the problem understanding solving ability.*
- ✓ *Design optimized solutions for a language.*

- Convert epsilon NFA to DFA.



- Minimize the following DFA.



- Construct Mealy and Moore Machines and equivalent them for the residue (remainder) mod 3 of binary input.
- Construct Finite Automata for the following Regular Expressions.
  a) RE=ab(a+b)*
  b) RE=(a+ab)(ab+ab)*
- Prove that the following languages are nor Regular.
  a) L={a^p | p is a prime number}
  b) L={b^n | n=i^2 and i>1}
  c) L={ WW^R | W is (a,b)*}
  d) L={a^nb^n+1 |n≥1}

MODULE-2

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

CONTEXT FREE GRAMMAR

Definition, Examples, Derivation, Derivation trees, Ambiguity in grammar, Inherent ambiguity, Ambiguous to unambiguous CFG, Useless symbols, Simplification of CFGs; Normal forms for CFGs - CNF and GNF, CFLs; Closure properties of Decision properties of CFLs-Emptiness, Finiteness and membership, pumping lemma for CFLs (proofs not Required), Application of pumping lemma.

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

PDA AND TM

Push Down PDA AND TM Automata (PDA): Description and definition, Instantaneous description, Language of PDA, Acceptance by final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA.

Turing Machines (TM): Basic model, Definition and representation, Instantaneous Description, Language acceptance by TM, Computable functions, Types of Turing Machines, Universal TM, Recursive and Recursively Enumerable Languages, undecidability, Church Turing Thesis, Universal Turing Machine, The universal and diagonalization languages, Reduction between languages and Rice's Theorem.

PRACTICES:

- Construct CFG for the following:
  a) L={a^nb^n|n>1}
  b) L={WW^R | W is (a,b)*}
  c) L={a^p | p is a prime}

- Derive the strings 10001 using left most derivation and right most derivation and parse tree by using the following grammar. And show that grammar is ambiguous.
  S-> T000T
  T->0T|1T|∈
- Convert the following CFG to CNF.
  S->ABC|Aa
  A->a
  B->d|∈
  C-> Aab|a
- Convert the following CFG to GNF.
  S->AA | 0
  A->SS | 1
- Prove that the following are not CFL.
  e) L={a^p | p is a prime number}
  f) L={b^n | n=i^2 and i>1}
  g) L={ WW^R | W is (a,b)*}
  h) L={a^nb^{n+1} |n≥1}
- Convert the following language or PDA to CFG.
  a) L={a^nb^n|n≥1} and
  b) δ(q,0,z)={(q,xz)}
  c) δ(q,0,x)={(q,xx)}
  d) δ(q,1,x)={(q,x)}
  e) δ(q, ∈ , x)={(p, ∈)}
  f) δ(p,1,x)={(p,xx)}
  g) δ(p,∈ , x)={(p, ∈)}
  h) δ(p, 1,z)={(p, ∈)}
- Construct PDA for the following Languages.
  a) L={0^n1m | n≥m}
  b) L={a^nb^n|n≥1}
  c) L= L={w\w ∈ {a, b}*
  d) $L=\{w|\ n_a(w) > n_b(w)\}$
  e) L={0^n1^{2n} | n>0}
  f) L=Where $w^R$is reverse of w
  g) L= Where $w^R$is reverse of w
  h) L= $\{wcw^R \mid w \in \{a,b\}^*\}$ Where $w^R$is reverse of w
- Construct PDA for the following Languages.
  a) L= {$a^nb^nc^n$ | n >1}
  b) L={$a^nb^ma^mb^n$ | n,m≥1}
- Construct Turing Machine for the following Languages.
  a) L={a^nb^n | n ≥ 1}
  b) L={$0^{2n}1^n$ |n >0 }
  c) L={ww^r | w is (0+1)*}
  d) L={$a^nb^nc^n$|n≥1}.
  e) Well balanced Parenthesis for example: ()()

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Design abstract models of computing, including Deterministic Finite Automata (DFA), non-deterministic Finite. Automata (NFA), Push Down Automata (PDA) and Turing Machine (TM) models and their power to recognize the languages.	Analyze	1,2	1,2,3
2	Design different finite state machines to perform various operations.	Apply	1,2	1,2,3
3	Analyze the given language is regular or not regular, CFL or not, Ambiguous unambiguous, Recursive and Recursive Enumerable.	Analyze	1,2	2
4	Design Regular grammar and context free grammars for a language.	Apply	1,2	1,3

TEXT BOOK:

1. Hopcroft and Ullman, “Introduction to Automata Theory, Languages and Computation”, 2nd Edition, Pearson/ Prentice Hall India, 2007.

REFERENCE BOOKS:

1. Zed A Shaw, Learn C the Hard Way: Practical Exercises on the Computational Subjects You Keep Avoiding (Like C), Addison Wesley, 2015.
2. Christoph Dürr, Sorbonne University, Jill-Jênn Vie, Inria, Competitive programming in Python, Cambridge University Press, 2020.
3. Michael Sipser, “Introduction to Theory of Computation”, 3rd Edition, Course Technology, 2012.

III

YEAR

COMPUTER SCIENCE AND ENGINEERING

B.Tech.

COURSE CONTENTS

I SEM & II SEM

I SEMESTER

22TP301	-	Soft Skills Laboratory
22CS301	-	Introduction to Artificial Intelligence
22CS302	-	Compiler Design
22CS303	-	Web Technologies
22CS304	-	Inter-Disciplinary Project - Phase - I
22CS305	-	Industry Interface Course
	-	Department Elective - 1
	-	NCC/NSS/SAC/E-Cell/Student Mentoring/Social Activities/Publication

II SEMESTER

22TP302	-	Quantitative Aptitude and Logical Reasoning
22CS204	-	Computer Networks
22CS306	-	Data Mining Techniques
22CS307	-	Software Engineering
22CS308	-	Inter-Disciplinary Project - Phase - II
	-	Department Elective - 2

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
0	0	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,

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*

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	Blooms Level	Module No.	Mapping with POs
1	Have the ability to introspect on individual strengths and weaknesses, and emerge as a balanced personality with improved self-awareness and self-worth.	Apply	1	12
2	Observe gender sensitive language and workplace etiquette in his professional life.	Analyze	1	9
3	Be able to prepare a resume and gain the confidence to face an interview.	Create	1, 2	10
4	Possess the interpersonal skills to conduct himself/herself effectively in everyday professional and social contexts.	Apply	2	8
5	Bring professionalism into his/her daily activities.	Create	2	8

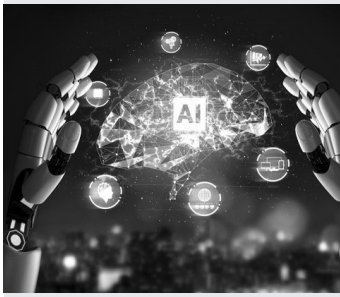
TEXT BOOKS:

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.
3. Krishna Mohan & NP Singh, "Speaking English effectively", Macmillan, 2008.
4. Rajiv K. Mishra, "Personality Development", Rupa & Co. 2004.

22CS301 INTRODUCTION TO ARTIFICIAL INTELLIGENCE



Source: https://www.forbes.com/sites/bernardmarr/2020/08/03/3-important-ways-artificial-intelligence-will-transform-your-business-and-turbocharge-success/

Hours Per Week :

L	T	P	C
2	2	0	3

PREREQUISITE KNOWLEDGE: 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 Agent, Agents and environments, Structure of agents, the concept of rationality, the nature of environments, Types of agents, problem solving approaches to typical AI problem.

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

PROBLEM SOLVING

State Space Problem; Searching: Uniform search, Informed Search: Solving problems by searching: Heuristic functions, Hill climbing, Best First Search, A* algorithm, AO* algorithm, Searching game trees: Min Max Search, Alpha Beta pruning.

PRACTICES:

- In the classical vacuum cleaner problem, we have two rooms and one vacuum cleaner. There is dirt in both the rooms and it is to be cleaned. The vacuum cleaner is present in any one of these rooms. Find the solution, how we can reach to reach a state in which both the rooms are clean and are dust free.
- In this problem, three missionaries and three cannibals must cross a river using a boat which can carry at most two people, under the constraint that, for both banks, that the missionaries present on the bank cannot be outnumbered by cannibals. The boat cannot cross the river by itself with no people on board. Find the solution, how to solve the given problem.
- You are given two jugs, a 4-gallon one and a 3-gallon one, a pump which has unlimited water which you can use to fill the jug, and the ground on which water may be poured. Neither jug has any measuring markings on it. Find the solution, how can you get exactly 2 gallons of water in the 4-gallon jug?
- There is a farmer who wishes to cross a river but he is not alone. He also has a goat, a wolf, and a cabbage along with him. There is only one boat available which can support the farmer and either of the goat, wolf or the cabbage. So at a time, the boat can have only two objects (farmer and one other). But the problem is, if the goat and wolf are left alone (either in the boat or onshore), the wolf will eat the goat. Similarly, if the Goat and cabbage are left alone, then goat will eat the cabbage. The farmer wants to cross the river with all three of his belongings: goat, wolf, and cabbage. What strategy he should use to do so?
- Either place a block that doesn't have other blocks stacked on top of it on another block with the same behaviour, or on the table. The initial and the goal state are described by the exact position of each block. Find the solution, how to solve the given problem.

SKILLS:

- ✓ *Analyze Intelligent systems.*
- ✓ *Apply problem solving techniques.*
- ✓ *Interface various knowledge representation.*
- ✓ *Create a dynamic planning.*

- Given a 3×3 board with 8 tiles (every tile has one number from 1 to 8) and one empty space. The objective is to place the numbers on tiles to match the final configuration using the empty space. We can slide four adjacent (left, right, above, and below) tiles into the empty space. Find the solution, how to solve the given problem by using using A* search algorithm.
- The rules of tic-tac-toe on the 3 × 3 field are as follows. Before the first turn all the field cells are empty. The two players take turns placing their signs into empty cells (the first player places Xs, the second player places Os). The player who places Xs goes first, the another one goes second. Find the solution, how to solve the given problem where the winner is the player who first gets three of his signs in a row next to each other (horizontal, vertical or diagonal).
- In crypt arithmetic problem, the digits (0-9) get substituted by some possible alphabets or symbols. The task in crypt arithmetic problem is to substitute each digit with an alphabet to get the result arithmetically correct. Find the solution, how to solve the given problem, where we can perform all the arithmetic operations on a given crypt arithmetic problem.

MODULE-2

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

KNOWLEDGE REPRESENTATION AND PLANNING

Propositional logic: Inference in propositional logic, Resolution, Forward chaining, Backward chaining, First order logic: Reasoning patterns in First order logic, Resolution, Forward chaining, Backward chaining, The planning problem: Planning with state space search, Partial order planning, Planning graphs.

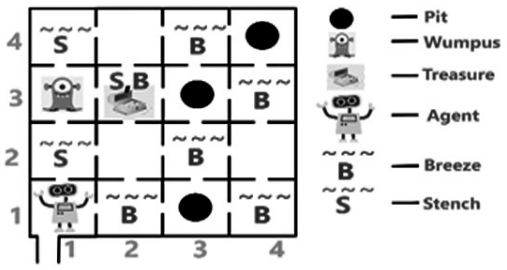
UNIT-2 **6L+6T+0P=12 hours**

LEARNING

Forms of learning: Supervised Learning, Unsupervised learning, Reinforcement learning, Learning Decision Trees, Ensemble Learning, Expert system.

PRACTICES:

- With logic programming, compare expressions and find out unknown values.
- The Wumpus world is a cave with 16 rooms (4×4). Each room is connected to others through walkways (no rooms are connected diagonally). The knowledge-based agent starts from Room [1, 1]. The cave has – some pits, a treasure and a beast named Wumpus. The Wumpus cannot move but eats the one who enters its room. If the agent enters the pit, it gets stuck there. The goal of the agent is to take the treasure and come out of the cave. The agent is rewarded, when the goal conditions are met. The agent is penalized, when it falls into a pit or being eaten by the Wumpus. Some elements support the agent to explore the cave, like -The Wumpus's adjacent rooms are stench. -The agent is given one arrow which it can use to kill the Wumpus when facing it (Wumpus screams when it is killed). – The adjacent rooms of the room with pits are filled with breeze. -The treasure room is always glittery. Find the Wumpus presented room.



- you are on one side of a river with a wolf, a goat, and a cabbage. You want to transport all three to the other side of the river, but you can only transport one object at a time. You cannot leave the wolf and the goat alone, or the cabbage and the goat alone; you are the only thing keeping them from eating each other. How can you transport everything from one side of the river to the other? Formulate it in terms of a Planning Domain Definition Language (PDDL).

- Impliment the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
- Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
- Implement k-nearest neighbors classification using python.
- Implement linear regression using python.
- Implement the naïve Bayesian classifier for a sample training dataset. Compute the accuracy of the classifier, considering few test data sets.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply AI search Models and Generic Search strategies for problem solving.	Apply	1	1,3
2	Inspect and analyze Logic for representing Knowledge and Reasoning of AI systems and Conduct investigation and implement project using AI learning techniques.	Analyze	1	2
3	Apply and evaluate the searching strategies to achieve the goal for a given situation..	Apply	2	6
4	Design different learning algorithms for improving the performance of AI systems.	Apply	2	4

TEXT BOOK:

1. S. Russel and P. Norvig, “Artificial Intelligence – A Modern Approach”, 4th Edition, Pearson Education, 2010.

REFERENCE BOOKS:

1. David Poole, Alan Mackworth, Randy Goebel, “Computational Intelligence : a logical approach”, Oxford University Press,2013.
2. G. Luger, “Artificial Intelligence: Structures and Strategies for complex problem solving”, 4th Edition, Pearson Education, 2008.
3. J. Nilsson, “Artificial Intelligence: A new Synthesis”, Elsevier Publishers. 2012.

Source: https://www.javatpoint.com/compiler-tutorial

22CS302 COMPILER DESIGN

Hours Per Week :

L	T	P	C
2	2	0	3

PREREQUISITE KNOWLEDGE: Programming for problem solving- I & II and Formal languages and automata theory.

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the foundation for understanding the theory and practice of compilers and compiler design concepts; symbol table management, compiler parsing techniques, semantic analysis and optimized code generation. This course introduced 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, code optimization and code generation.

MODULE-1

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

INTRODUCTION

The evolution of programming languages and basic language processing system; The structure of a compiler; Bootstrapping; Lexical analyser and its Role; Input buffering; Specifications and recognition of tokens; LEX.

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

SYNTAX ANALYSIS

The role of the parser; Context-free grammars; Types of parsers with examples, YACC.

Semantic Analysis: Type checking; Syntax directed definition (SDD) and translation schemes (TS); Application of SDD and TS; Translation of expressions and control flow statements.

PRACTICES:

- Implement various phases of compiler in detail. Write down the output of each phase for expression Total = (b + c) + (b + c) * 50.
- Construct the symbol table for any input file with the help of LEX tool.
- Consider the context free grammar.
  S→SS+, S→SS*, S→a and the string aa+a*.
  i) Give the leftmost derivation for the string.
  ii) Give the rightmost derivation of the string.
  iii) Is the grammar ambiguous or not.
- Check whether the following grammar is a LL (1) grammar.
  S →iEtS | iEtSeS | a, E → b.
- Construct the FIRST and FOLLOW procedures for the following grammar.
  S → Aa | bAC | dc | bda, A → d.
- Consider the grammar,
  E→TE', E'→+TE' | €, T → FT', T'→ *FT' | €, F → (E) | id.
  Construct a predictive parsing table for the grammar given above. Verify whetherthe input string id + id * id is accepted by the grammar or not.

MODULE-2

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

INTERMEDIATE REPRESENTATIONS

Three address code; Syntax tree; DAG.

Run-Time Environment: Storage organization; Stack allocation - Activation Trees, Activation Records.

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

OPTIMIZATION AND CODE GENERATION

The principal sources of optimization; Basic blocks and flow graphs; Local optimization; Global optimization and loop optimization.

Code Generation: Issues in the design of code generator; Code-generation algorithm – register allocation and assignment and peephole optimization.

PRACTICES:

- Translate the executable statements of the following C-code segment into three address code.

```
int i:
int a[10]
i = 0;
While (I <= 10) {
a[i] = i + 1; i + + ;
}
```

- Compute the DAG for the following three address statements. Considering this DAG as an example, explain the process of code generation from DAG.

  t1 = a + b t2 = c + d t3 = e – t2 t4 = t1 - t3

- What is Data flow equation? Represent the Data flow information for the following

  a = b + c; d = c * d; e = a – c; f = d + e.

- Draw a flow graph for the below code. Show the basic blocks clearly in your control flow graph?

```
If ( i&gt;=0){
sum = B[0];
i = 0;
L1: if (A[i]&lt; B[i]){
j=i;
L2:
if( B[i]&gt;=0){
sum = sum +B[j];
}
j = j+1
if ( j&lt;N) goto L2;
}
i = i+1
if ( i&lt;N) goto L1;
}
```

SKILLS:

- ✓ *Design parsers using top-down and bottom-up approaches.*
- ✓ *Usage of tools like LEX and YACC.*
- ✓ *Design a simple code generation*

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply the different phases of compiler with various examples.	Apply	1	1, 12
2	Design different parsing and optimization techniques in the design of compile.	Design	1	1, 2, 12
3	Analyze the code optimization techniques.	Analyze	2	1, 2, 3, 12
4	Analyze the algorithm for compiler segments and evaluate the algorithm for optimized code generation.	Analyze	2	1, 2, 3,12

TEXT BOOKS:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ulman, “Compilers: Principles, Techniques and Tools”, 3rd Edition, Pearson Education, 2019.
2. Thomson, “Introduction to Theory of Computation”, 2nd Edition, Sipser, 2016.

REFERENCE BOOKS:

1. V. Raghavan, “Principles of Compiler Design”, 2nd Edition, Mc Graw Hill, 2016.
2. John R.Levin, Tony Mason and Doug Brown, “Lex & YAAC”, 2nd Edition, O Reilly, 2012.
3. Ms. Manisha Bharambe, “Compiler Construction”, 2nd Edition, Nirali Prakashan, 2017.

22CS303 WEB TECHNOLOGIES

Source: https://www.dreamstime.com/web-development-coding-programming-internet-technology-business-concept-web-development-coding-programming-internet-technology-image121903546

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: OOPs through JAVA.

COURSE DESCRIPTION AND OBJECTIVES:

This course offers the concepts of web development like static and dynamic web page design and provides internet programming knowledge, web servers, application servers, and design methodologies using object-oriented concepts. The objective of this course is to build web applications using JSP, PHP, Angular JS, and Node JS with a client and server-side scripting technologies that span multiple domains.

MODULE - 1

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

INTRODUCTION

HTML: Creating structured documents, Links and navigation, Tables, Forms, and Frames.

HTML 5: Introduction to HTML5, The HTML5 Canvas, HTML5 audio and Video;

CSS: Cascading Style Sheets, CSS Properties.

Java Script: Learning Java Script- how to add scripts to your page, DOM, variables, operators, functions, conditional statements, Looping, Events, Built-in objects, form and regular expression validation.

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

JDBC AND JSP

JDBC: What is JDBC, system requirements, types of JDBC Drivers, creating database tables, connecting to a database, executing SQL statements, processing result sets, and making changes to a result set.

JSP: JSP Processing, Generating Dynamic Content using Scripting Elements, Implicit JSP Objects, Sharing Data between JSP pages, JSP application design with JDBC.

PRACTICES:

- Design a webpage having four frames named a)Top, b)Center, c)Bottom, and d) Left. The top frame should contain the company logo and title. The bottom frame should contain copyright information. The left frame should contain various links like Home, Products, Services, Branches, About, etc., When clicked on respective links, the content should display on the center frame.
- Design a catalog page that should contain the details of all the books available on the website in a table. The details should contain the following: a) Snapshot of Cover Page b) Author Name c) Publisher. d) Price. e) Add to cart button.
- Design a timetable schedule for your current semester using the Table tag.
- Design a HTML page for Student Registration Form using Form ElementsthatincludesForm,input-text,password,radio,checkbox,hidden,button,submit,reset,label,textarea,select,option and file upload.
- Design a HTML web page with at least two <h1>, two images, two buttons, and appropriate CSS to display,

  All<h1>withfont-size12pt, and bold in Verdana font using In line CSS.

  All<img>withbordercoloryellow,thickness10pxusingDocumentLevelCSS.

  All<inputtype='button'>shouldchangebackgroundcolortoredonmouseoverthemusing External CSS.

SKILLS:

- ✓ *Perform client-side validation using Java Script and Angular JS.*
- ✓ *Store and retrieve data using Node JS.*
- ✓ *Generate dynamic web pages using JSP and PHP.*
- ✓ *Develop a web application or website for any real-time requirements.*

- 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:

  Factorial of that number.

  Fibonacci series up to that number.

  Prime numbers up to that number.

  Is it palindrome or not?
- Design a web page that contains a color pallet, when the user moves the mouse to the particular area, then it changes the background color of the web page.
- Design a registration page to validate the following fields using Java Script.

  Make sure the user name starts with an upper case letter.

  The user name must have at least one digit.

  Ensure that Email is valid.

  Ensure that the password length is between 8 to 20 characters.

  Make sure the password contains at least one upper case letter, one lower case, and one special character exclude [. (dot), ,(comma), ;(semicolon), : (colon)].
- Design a web page to display the videos on-page, on user selection using frames and HTML5 tags.
- Design a web page to display different types of objects using HTML5 Canvas.
- Design a web application to validate entered username and password through JDBC connection program and display user information on successful login and provide profile editing option to the user. Else display an error message.
- Develop a JSP application to create a user on successful signup and update user information on successful login and display user information on the home screen and provide a logout button.
- Make an HTML form that collects the last name. Send the name to JSP page. If there is an employee with that last name, show full details of him or her (just show the first employee if there are multiple people with the same name). If there is no employee with that last name, say “no employee records available.”

MODULE-2

UNIT–1 **8L+0T+8P=16 Hours**

PHP

Introduction to PHP, Expressions, and control flow in PHP, functions and objects, Arrays, Accessing MySQL using PHP, Form Handling, Cookies, Sessions, and Authentication.

UNIT–2 **8L+0T+8P=16 Hours**

ANGULAR JS AND NODE JS

Angular JS: Introduction, Expressions, Modules, Directives, Controllers, Filters, Events, Forms, Form Validation.

Node JS: Introduction, Setup Dev Environment, Modules, Node Package Manager, Creating Web server, File System, Events, Express.js, Accessing MySQL from Node.js.

PRACTICES:

- Design a web page using PHP, upload image into web page and display image, when user clicking on view button.
- Design a personal Information form, Submit & Retrieve the form data using $_GET(), $_POST() and $_REQUEST() Variables.
- Design a login page to validate username and password through MySQL. If login is successful display user information on home page and modify user information on edit page using sessions. When user logged out, destroy all user-related sessions.
- Design a web page to accept payment data from user and do the payment, on successful

payment display details on the screen. A Session should be set while doing payment up to 10 minutes after that link/payment page should be destroyed irrespective of user payment.

- Design a web application to validate user registration page using Angular JS.
- Design a search engine using Angular JS. On key press, display data on web page.
- Design a web page to validate user name and password using: Node.js and PHP. When user clicks on login button, server checks the data availability in data base. If the data matches a successful login page is returned. Otherwise, a failure message is shown to the user.
- Design a web application to display the active duration of the user, i.e., time between login and logout.

COURSE OUTCOMES:

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

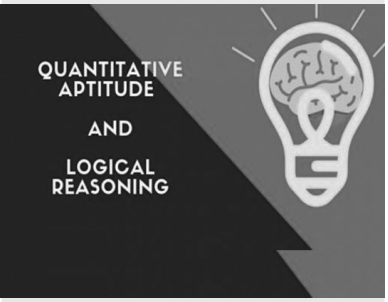
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Usage of HTML, HTML5, CSS, Java Script, and PHP in web application development.	Apply	1, 2	1
2	Apply Angular JS features for form validation and Node JS, and JDBC concepts to perform database operations from web pages.	Apply	1, 2	1
3	Analyse the suitability of Node JS and JSP technologies to build solutions for real-world problems.	Analyse	2	2
4	Design and develop three tier web applications using JSP, Node JS, Angular JS, and PHP.	Creating	2	3

TEXT BOOKS:

1. Jon Duckett, “Beginning Web Programming with HTML, XHTML, and CSS”, 2nd Edition, Wiley Publishing, Inc, 2008.
2. Robin Nixon, “Learning PHP, MySQL & JavaScript WITH JQUERY, CSS & HTML5”, 4th Edition, O’Reilly, 2015.

REFERENCEBOOKS:

1. Paul Deitel, Harvey Deitel, Abbey Deitel, “Internet & World Wide Web How to Program”, 5th Edition, Pearson Education, 2012.
2. Kishori Sharon, “Java APIs, Extensions and Libraries with JavaFX, JDBC, jmod, jlink, Networking and the process API”, 2nd Edition, Apress, 2018.
3. Brad Dayley, Brendan Dayley, and Caleb Dayley, “Node.js, Mongo DB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications,” 2nd Edition, Pearson Education, 2018.
4. Steve Prettyman, “Learn PHP 7 Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL”, 1st edition, Apress, 2015.
5. Adrian W. West and Steve Prettyman, “Practical PHP 7, MySQL 8, and MariaDB Website Databases: A Simplified Approach to Developing Database-Driven Websites”, 1st edition, A Press, 2018.



Source: https://images.app.goo.gl/kvtVgA8TkvDCqLhj7

22TP302 QUANTITATIVE APTITUDE AND LOGICAL REASONING

Hours Per Week :

L	T	P	C
1	2	0	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 a 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 homework assignments in each concept.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Meet the demands of current job market besides equipping them higher studies like CAT, GMAT etc.	Apply	1	2, 5
2	Solve Arithmetic and Reasoning Problems within shortest possible time without paper work.	Apply	1	2, 5
3	Exhibit better analytical skills and aptitude skills.	Analyse	2	2, 4
4	Develop interpretational skills.	Evaluation	2	2, 4

SKILLS:

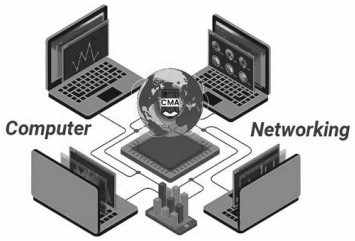
- ✓ *Helps in developing and improving problem-solving skills.*
- ✓ *Flexing and honing logical abilities.*
- ✓ *Allow students to develop critical thinking skills.*

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, 2013.
2. R. S. Aggarwal, “A Modern Approach to Verbal & Non-Verbal Reasoning”, Revised Edition, S. CHAND Publications, 2018.



Source: https://snabaynetworking.com/what-is-computer-network-and-its-types/

22CS204 COMPUTER NETWORKS

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: JAVA programming and UNIX commands.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on imparting knowledge about various protocols involved in LANs and WANs. In addition, it gives a good foundation on different protocols such as data link protocols, internet protocols, and transport protocols present in the respective layers of the data communication system.

MODULE-1

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

INTRODUCTION TO COMPUTER NETWORKS AND INTERNET

Understanding of network and Internet, the network edge, the network core, Understanding of Delay, Loss and Throughput in the packet switching network, protocols layers and their service model, History of the computer network.

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

APPLICATION LAYER & TRANSPORT LAYER

Principles of computer applications, Web and HTTP, E-mail, DNS, Socket programming with TCP and UDP.

Introduction and transport layer services, Multiplexing and Demultiplexing, Connectionless transport (UDP), Principles of reliable data transfer, Connection-oriented transport (TCP), Congestion control.

PRACTICES:

- Install Network Simulator 2/3. Create a wired network using dumbbell topology. Attach agents, generate both FTP and CBR traffic, and transmit the traffic. Vary the data rates and evaluate the performance using metric throughput, delay, jitter and packet loss.
- Create a static wireless network. Attach agents, generate both FTP and CBR traffic, and transmit the traffic. Vary the data rates and evaluate the performance using metric throughput, delay, jitter and packet loss.
- Create a mobile wireless network. Attach agents, generate both FTP and CBR traffic, and transmit the traffic. Vary the data rates and evaluate the performance using metric throughput, delay, jitter and packet loss.
- Implementation of one-way and two-way communication using TCP / UDP.
- Hello command is used to know whether the machine at the other end is working or not. Echo command is used to measure the round trip time to the neighbour. Implement Hello and Echo commands using JAVA.

MODULE-2

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

NETWORK LAYER

Introduction to forwarding and routing, Network Service models, Virtual and Datagram networks, study of router, IP protocol and addressing in the Internet, Routing algorithms, Broadcast and Multicast routing.

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

THE LINK LAYER AND LOCAL AREA NETWORKS

Introduction to link layer services, error detection, and correction techniques, Multiple access protocols, addressing, Ethernet, switches, and VLANs.

PRACTICES:

- Find all the IP addresses on your network using Unicast, Multicast, and Broadcast on your network.
- Use Packet tracer software to build network topology and configure using Distance vector routing and Link State routing protocols.
- Study different types of Network cables (Copper and Fiber) and prepare cables (Straight and Cross) to connect two or more systems.
  a. Use a crimping tool to connect jacks.
  b. Use a LAN tester to connect the cables.
  c. Install and configure Network Devices: HUB, Switch and Routers (Consider both manageable and non-manageable switches. Perform logical configuration of the system and set the bandwidth of different ports).
  d. Install and Configure Wired and Wireless NIC and transfer files between systems in Wired LAN and Wireless LAN. Consider both ad-hoc and infrastructure modes of operation.
- Apply the commands such as Ping, Tracert, Ipconfig, pathping, telnet, FTP, getmac, ARP, Hostname, Nbtstat, netdiag, and Nslookup to solve various problems.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Build the basic concepts of Network hardware, software and reference models.	Apply	1	1, 2, 12
2	Evaluate different physical layer media and switching methods.	Evaluation	1	1, 2, 5, 12
3	Implement various protocols with modern tools.	Apply	1	1, 2, 3, 5, 12
4	Apply different protocols to perform end-to-end delivery and interaction with users.	Analyze	2	1, 2, 12
5	Analyze various design issues, protocols and functionalities of network layer.	Analyze	2	1, 2, 12
6	Demonstrate various protocols involved in data link layer operations.	Apply	2	1,2, 5

TEXT BOOKS:

1. Kurose and Ross, “Computer Networking- A Top-Down approach”, 6th Edition, Pearson, 2017.
2. Behrouz Forouzan, “Computer Networks- A Top-Down approach”, McGraw Hill, 2014.

REFERENCE BOOKS:

1. Andrew S. Tanenbaum, “Computer Networks”, 5th edition. Pearson Education, 2014.
2. Behrouz A. Forouzan, “Data communications and Networking”, 5th edition, TMH, 2017.
3. William Stallings, “Data and Computer Communications”, 10th edition, Pearson Education, 2017.
4. Fred Halsall, “Computer Networking and the Internet”, 5th edition, Addison Wesley, 2005.

SKILLS:

- ✓ *Establish local area networks with different topologies.*
- ✓ *Design of new routing protocols.*
- ✓ *Network troubleshooting such as installing network interface card drivers, setting IP addresses, subnet masking, etc.*

Source: https://alternative-spaces.com/blog/8-data-mining-techniques-you-must-learn-to-succeed-in-business/

22CS306 DATA MINING TECHNIQUES

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Probability and statistics, Python programming.

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the basic concepts, principles, methods, implementation techniques, and applications of data mining, with a focus on three major data mining functions: (1) Association rule mining (2) Classification and (3) cluster Analysis. It also focuses on issues relating to the feasibility, usefulness, effectiveness and scalability of techniques for the discovery of patterns hidden in large data sets.

MODULE-1

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

INTRODUCTION

What is Data Mining? Why data mining?; What kinds of data can be mined?;What kinds of patterns can be mined?; Which technologies are used?; What kinds of applications are targeted? Major issues in data mining; Data objects and attribute types; Basic statistical descriptions of data, Data matrix versus dissimilarity matrix.

Data Pre-processing: Overview - data quality, major tasks in data preprocessing; Data cleaning - missing values, noisy data; Data Integration - entity identification problem, redundancy and correlation analysis tuple duplication; Data value conflict detection and resolution; Data reduction - PCA, attribute subset selection, regression and log linear models; Histogram; Data transformation - data transformation by normalization; Discretization by binning;

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

ASSOCIATION ANALYSIS

Market basket analysis; Frequent Item sets; Closed item sets and association rules; Frequent Item set Mining Methods-apriori algorithm, generating association rules, improving apriori, FP growth method, vertical format method; Which patterns are interesting? Pattern evaluation method; Pattern Mining in multilevel multidimensional space, Pattern Mining in Multilevel, Multidimensional Space.

PRACTICES:

- Apply the following data pre-processing techniques on dataset (download from n UCI/ Kaggle/
- NCBI data repository) to illustrate the need of the pre-processing in data mining
  a) Data Cleaning
  b) Data Normalization
  c) Data Discretization
  d) Computation of correlation coefficient to analyze the data behavior
  e) Dimensionality reduction using PCA and Wavelets
- Construct Heat Map Table to understand the Correlation among the attributes in a given dataset.
- Extract the interesting association rules from a given dataset using A priori algorithm.
- Extract the interesting association rules from a given dataset using Frequent Pattern growth algorithm.

MODULE-2

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

CLASSIFICATION

What is classification?, General approach to classification, Decision tree induction - attribute selection measures; Tree pruning; Bayes Classification methods - Bayes theorem; Naïve Bayesian classification; Classification by back propagation - a multilayer feed forward neural network; Defining a network topology; Back propagation; K nearest neighbor classifier; Support vector machine, Linearly separable and inseparable cases, Model evaluation and selection; Techniques to improve classification accuracy.

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

CLUSTER ANALYSIS

Partition methods - K means and K medoid; Hierarchical methods; Agglomerative and divisive method; Density based methods - DBSCAN; Optics; Grid based methods-STING; Cluster evaluation methods; Clustering high dimensional data; Problems, Challenges and major methodologies

PRACTICES:

- Apply the following classifiers on a given dataset and analyze their performance.
  a) J48 and visualize the decision tree.
  b) Naive Bayes.
  c) Support Vector Machine.
  d) Multi-Layer Perceptron.
  e) K-Nearest Neighbor.
- Illustrate the performance of Ensemble Classification algorithms such as Bagging and Boosting Methods.
- Evaluate the performance of partitioning clustering algorithms on a given dataset.
- Evaluate the performance of hierarchical clustering algorithms on a given dataset.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Investigate various patterns that can be extracted from different types of data.	Analyze	1,2	1, 2,
2	Apply various pre-processing techniques and classification algorithms on different domains of data.	Apply	1,2	1, 2, 5, 6
3	Build decision making systems using data mining algorithms for a given real time data set.	Apply Create	1,2	3, 5, 8
4	Construct models using modern tools such as WEKA, R and Python etc.	Apply Create	1,2	1, 2,5,9

TEXT BOOKS:

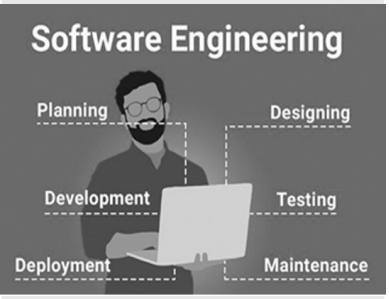
1. Jiawei Han, Micheline Kamber and Jian Pei, “Data mining Concepts and Techniques”, 3rd Edition, Morgan Kaufmann. 2012.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction to Data Mining”, 2nd Edition, Pearson, 2018.

REFERENCE BOOKS:

1. Jure Leskovec, Anand Raja raman and Jeffrey D Ullman, “Mining of Massive Datasets”, 5th Edition, Stanford University, 2014.
2. GK Gupta, Introduction to Data Mining with Case Studies, Prentice Hall. 3rd Edition,2014.
3. Margaret H Dunham, “Data Mining: Introductory and Advanced Topics”, PEA, 2008.

SKILLS:

- ✓ *Handle various types of Data and able to explore the characteristics of data*
- ✓ *Perform various Data Visualisation tasks over the data and present the data with ease of access*
- ✓ *Perform descriptive and predictive mining tasks over the data to carry out decision making.*



Source: https://artoftesting.com/software-engineering

22CS307 SOFTWARE ENGINEERING

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Data base management systems, OOPs through Java.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the concepts of software 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 efficient, cost effective, feasible software as per user requirements.

MODULE - 1

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

INTRODUCTION

Introduction to Software Engineering: Introduction to Software and Software engineering, Software characteristics, Software project, Software myths, Project Planning, Scheduling and Management.

Generic View of Process: Software Engineering - A layered technology, A process framework, Software Development Life Cycle (SDLC), The Capability Maturity Model Integration (CMMI).

Process Models: Conventional Model, Agile process models - Unified process model, Extreme Programming, Scrum.

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

REQUIREMENTS ENGINEERING

Requirements Engineering: Functional and Non-functional requirements, User requirements, System requirements, Requirement engineering tasks, formal requirements specification and verification, Feasibility Study.

Building the Analysis Model: Data modeling - Data objects, Attributes, Relationships, Cardinality and modality. Class based modeling - Identify analysis classes, specify attributes and Define operations.

Design Engineering: Design model, Design concepts. Creating an Architectural Design-Architectural styles and patterns.

Performing User Interface Design: Golden rules; User interface analysis and design.

PRACTICES:

Laboratory session of this course is designed in such a way that the student should complete three projects of the given type by performing the below experiments.

- Development of software requirements specification using Mind-Map tool.
- Project planning using Gantt charts.
- Project estimation using metrics.
- Capture Use Case Scenarios and model UML Use Case Diagrams.
- Model the UML state chart and Activity diagrams.
- Model the UML Class and Sequence diagrams.

MODULE - 2

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

TESTING

Testing Strategies: A strategic approach to software testing, Unit testing, Integration testing, Validation testing, System testing,

Testing Tactics: Black-Box and White-Box testing techniques, Art of debugging.

Product Metrics: Metrics for analysis model; Metrics for design model, Metrics for source code; Metrics for testing; Metrics for maintenance.

UNIT–2 **8L+0T+8P=16 Hours**

RISK AND QUALITY MANAGEMENT

Risk Management: Software risks, Risk identification; Risk projection; Risk refinement, Reactive vs Proactive risk strategies, RMMM.

Quality Management: Quality concepts, Formal technical reviews, Statistical Software Quality Assurance.

Computer-Aided Software Engineering (CASE): Use of appropriate CASE tools- Requirement engineering tools, Project planning tools, Testing tools.

PRACTICES:

Laboratory session of this course is designed in such a way that the student should complete three projects of the given type by performing the below experiments.

- Estimate the test coverage and Structural complexity of product using metrics.
- Develop the test cases for all the functional requirements of projects selected.
- Perform the functional testing using Selenium tool.

LIST OF PROJECTS:

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. Glasscode’s goal is to build that system.

Project-6: 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 several 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.

SKILLS:

- ✓ *Know the software requirements and find out various ways to gather and specify them.*
- ✓ *Choose a process model for developing software solutions without schedule/ effort overruns and good quality.*
- ✓ *Analyse and model (diagrammatical representations) a software product.*

Project-8: A Notes And File Management System: During 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 distrib- VFSTR 106 III Year I Semester uted 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

Project-11: Draft Software Requirement Analysis for the following Problem Statement: Fuel Delivery System: An unattended petrol (gas) pump system that includes a credit card reader. The customer swipes the card through the reader and then specifies the amount of fuel required. The fuel is delivered, and the customer's account debited.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Use basic concepts of software engineering for designing software product.	Usage	1	1, 11
2	Compare different process models and identify appropriate process model based on project requirements.	Evaluation	1	2, 4
3	Build Software Requirement Specification (SRS) document for any software product.	Design	1	3, 5
4	Design of solutions using UML diagrams like Use case, Sequence diagrams etc.	Design	1	3, 4, 5
5	Create an appropriate architecture for a given project that meets all quality constraints.	Create	2	5
6	Apply different testing techniques to ensure bug free software and metrics to measure the software size, complexity, and budget etc.	Apply	2	4, 5, 11

TEXT BOOKS:

1. Roger S. Pressman, "Software Engineering, A practitioner's Approach", 6th Edition, McGrawHill International Edition, 2008.
2. Booch G., Rumbaugh J. and Jacobsons I, "The Unified Modeling Language User Guide", 2nd Edition, Addison Wesley, 2005.

REFERENCE BOOKS:

1. Simon Sennet, Steve McRobb and Ray Farmer, "Object Oriented Systems Analysis and Design, 2nd edition, 2004.
2. Dr. Pankaj Jalote "Software Engineering: A Precise Approach"-edition 2010.

IV

YEAR

COMPUTER SCIENCE AND ENGINEERING

B.Tech.

I SEMESTER

22CS401	-	Cryptography and Network Security
22CS402	-	Big Data and Analytics
22CS403	-	Cloud Computing
	-	Department Elective - 3
	-	Department Elective - 4

II SEMESTER

22CS404	-	Project Work

COURSE CONTENTS

I SEM & II SEM

22CS401 CRYPTOGRAPHY AND NETWORK SECURITY

Source: https://www.brainkart.com/subject/CRYPTOGRAPHY-AND-NETWORK-SECURITY-PRINCIPLES-AND-PRACTICE_136/

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Computer networks.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the modern concepts of network security using various cryptographic algorithms and underlying network security applications. It enables to understand various symmetric and asymmetric cryptographic techniques. It focuses on providing security services in applications such as e-mail functioning, web security and secure electronic transactions protocol and system security.

MODULE-1

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

INTRODUCTION

Introduction To Computer and Network Security Concepts: Computer Security Concepts, Security attacks, Security services, Security mechanisms, Fundamental Security Design Principles, Attack Surfaces and Attack trees, A model for network security.

Classical Encryption Techniques: Symmetric cipher model, Substitution techniques, Transposition techniques

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

SYMMETRIC AND ASYMMETRIC CRYPTOGRAPHY

Symmetric Ciphers: Block cipher principles, Data encryption standard, Strength of DES, Blockcipher design principles, AES cipher, Multiple encryption and triple DES, Block cipher modes ofoperation, RC4.

Asymmetric Ciphers and Cryptographic Hash Functions: Principles of public keycryptosystems, RSA algorithm, Diffie-Hellman Key Exchange, Message Authentication requirements,Authentication functions, Message authentication Codes, Hash functions, Security of hash functionsand MACs, Digital signature standard.

PRACTICES:

- Implement Substitution and Transposition Ciphers
  - Ceaser cipher
  - Playfair cipher
  - Hill cipher
  - Rail fence cipher
- Implement Symmetric Cipher
  - S-DES
  - RC4
- Implement Asymmetric Cipher
  - RSA
  - Diffie-Hellman
  - Hash Function

SKILLS:

- ✓ *Design various security services for appropriate applications.*
- ✓ *Identifying the appropriate firewall, password management and anti-virus models for specific applications.*
- ✓ *Test and resolve threats and malfunctions in network.*
- ✓ *Apply different security mechanisms for web applications.*
- ✓ *Build authentication system for security protocols.*

MODULE-2

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

SECURITY APPLICATIONS

Network Security Applications: Kerberos, X.509 authentication service, Public key infrastructure,

E-Mail Security: Pretty good privacy, S/MIME.

IP Security Overview: IP security architecture, Authentication header, Encapsulating security payload, Combining security associations, key management.

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

WEB AND SYSTEM SECURITY

Web Security: Secure socket layer and transport layer security, HTTPS, Secure Shell (SSH)

System Security: Intruders, Intrusion detection, Malicious software, Firewalls

PRACTICES:

- Configure IP Address in a system in LAN (TCP/IP Configuration)
- Configure DNS to establish interconnection between systems
- Configuring Windows Firewall
- Adding users, setting permissions
- Configure Mail server
- Demonstrate the usage of Wireshark to identify abnormal activity in network communication.
- Demonstrate usage of NMAP (Zenmap) Tool in Network Scanning.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply cryptographic techniques in various security service solutions effectively in everyday professional and social contexts.	Apply	1,2	1,2
2	Analyze the usage of secure protocols to safeguard sensitive data using internet.	Analyze	1,2	1,2
3	Usage of tools to Identify abnormal activity in network communication to take appropriate action.	Apply	2	5
4	Apply various security protocols to safe guard the data internet using SSL/TCL.	Apply	2	1,2

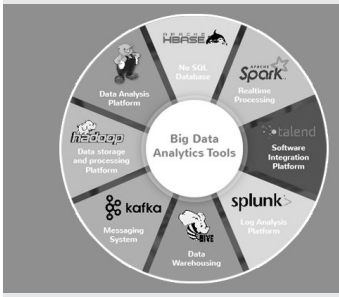
TEXT BOOK:

1. William Stallings, “Cryptography and Network security”, 7th Edition, Pearson Education, 2017.

REFERENCE BOOKS:

1. William Stallings “Network Security Essentials Applications and Standards”, 2nd Edition, Pearson Education, 2009.
2. Eric Malwald, “Fundamentals of Network Security”, 4th Edition, Pearson Education, 2010.
3. Buchmann, “Introduction to Cryptography”, 2nd Edition, Pearson Education, 2009.
4. Charlie Kaufman, “Radis Perlman and Mike Speciner, Network Security - Private Communication in a Public World”, 1st Edition, Pearson Education, 2009.

22CS402 BIG DATA AND ANALYTICS



Source: https://miro.medium.com/max/844/0*ARAg3FnAzy2e02Wy.png

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Basics of databases, Data mining.

COURSE DESCRIPTION AND OBJECTIVES:

This course serves as an introductory course to gain knowledge on analyzing Big Data. Expecting to face Big Data storage, processing, analysis, visualization, and application issues on both workplaces and research environments. Get insight on what tools, algorithms, and platforms to use on which types of real world use cases.

MODULE-1

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

INTRODUCTION TO BIG DATA

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, Challenges of big data.

Big data analytics: Overview of business intelligence, Data science and Analytics, Meaning and Characteristics of big data analytics, Need of big data analytics, Classification of analytics, Challenges to big data analytics, Importance of big data analytics, Basic terminologies in big data environment.

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

INTRODUCTION TO HADOOP

Introducing Hadoop, need of Hadoop, limitations of RDBMS, RDBMS versus Hadoop, Distributed Computing Challenges, History of Hadoop, Hadoop Overview, Use Case of Hadoop, Hadoop Distributors, HDFS (Hadoop Distributed File System), Processing Data with Hadoop, Managing Resources and Applications with Hadoop YARN (Yet another Resource Negotiator), Interacting with Hadoop Ecosystem.

PRACTICES:

- Hadoop installation in standalone machine.
- Pig installation.
- Setup of Hadoop cluster.
- HDFS basic command-line file operations.
- HDFS monitoring User Interface.

MODULE-2

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

MAPREDUCE PROGRAMMING

Introduction, Mapper, Reducer, Combiner, Partitioner, Searching, Sorting, Compression, Real time applications using Map Reduce, combiner, Partitioner, matrix multiplication using Map Reduce and page rank algorithm using Map Reduce.

SKILLS:

- ✓ *Build and maintain reliable, scalable, distributed systems with Apache Hadoop.*
- ✓ *Develop Map Reduce based applications for Big data.*
- ✓ *Design and build applications using Hive and pig based Big data applications.*
- ✓ *Learn tips and tricks for big data use cases and solutions.*

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

PIG

Introduction to Pig, The Anatomy of Pig, Pig on Hadoop, Pig Philosophy, Use Case for Pig: ETL Processing, Pig Latin Overview, Data Types in Pig, Running Pig, Execution Modes of Pig, HDFS Commands, Relational Operators, Piggy Bank, Word Count Example using Pig, Pig at Yahoo!.

Hive: Introduction to Hive, Hive Architecture, Hive Data Types, Hive File Format, Hive Query Language (HQL), Partitions and bucketing, working with XML files, User-Defined Function (UDF) in Hive, Pig versus Hive.

Spark Programming: Introduction, features of Spark, components of Spark, Programming with Resilient Distributed datasets (RDDS).

PRACTICES:

- Word Count Map Reduce program using Hadoop.
- Implementation of word count with combiner Map Reduce program.
- Practice on Map Reduce Monitoring User Interface.
- Implementation of Sort operation using Map Reduce.
- Map Reduce program to count the occurrence of similar words in a file by using partitioner.
- Design Map Reduce solution to find the years whose average sales is greater than 30.
  - input file format has year, sales of all months and average sales.
  - Year Jan Feb Mar April May Jun July Aug Sep Oct Nov Dec Average.
- Map Reduce program to find Dept wise salary.
  - Empno Emp Name Dept Salary.
- Designing of Pig Latin scripts to sort, group, join, project and filter the data.
- Implementation of Word count using Pig.
- Creation of Database and tables using Hive query language.
- Implementation of partitions and buckets using Hive query language.
- Implementation of word count using spark RDD.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Use of Big data frameworks like Hadoop and NOSQL to efficiently store and process Big data to generate analytics.	Apply	1	1, 2, 5,9,10,12
2	Design a solution for data intensive problems using Map Reduce paradigm.	Apply	1	1, 2, 5, 9,10,12
3	Design and analyze the solutions of Big data using Pig and Hive to solve data intensive and to generate analytics.	Apply	2	1, 2, 3, 5, 9,10,12
4	Analyze Big data using Spark programming.	Analyze	2	1, 2, 3, 5, 9,10,12

TEXT BOOKS:

1. Seema Acharya, Subhashini Chellappan, “Big Data Analytics”, Wiley, 2015.
2. Holden Karau, Andy Konwinski, Patrick Wendell, MateiZaharia, “Learning Spark: Lightning-Fast Data Analysis”, O’Reilly Media, Inc., 2015.

REFERENCE BOOKS:

1. Boris Lublinsky, KevinT. Smith, Alexey Yakubovich, “Professional Hadoop Solutions”, Wiley, 2015.
2. Chris Eaton,Dirk deRooset al. , “Understanding Big data ”, McGraw Hill, 2012.
3. Tom White, “HADOOP: The definitive Guide”, O Reilly 2012.

22CS403 CLOUD COMPUTING

Source: https://tse4.mm.bing.net/th/id/OIP.vN86lZCAdr3RDSXc0cuHcAHaE8?pid=ImgDet&rs=1

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Operating systems and Computer networks.

COURSE DESCRIPTION AND OBJECTIVES:

This course deals with the computing model, which enables information, software, and shared resources to be provisioned over the network as services in an on-demand manner. The main objective of this course is to enable the student to understand the evolution of cloud computing through its supporting technologies virtualization and the architectures of top cloud platforms.

MODULE-1

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

INTRODUCTION

Introduction: Definition, Historical developments, Computing platforms, and technologies.

Principles Of Parallel and Distributed Computing: Parallel versus distributed computing, Elements of parallel computing, Elements of distributed computing, Technologies for distributed computing.

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

VIRTUALIZATION

Introduction, Characteristics, Virtualization techniques, Virtualization and cloud computing, Pros. and cons. of virtualization, Technology examples.

Cloud Computing Architecture: Introduction, Cloud reference model, Types of clouds, Economics of clouds, Open challenges.

Cloud Platforms in Industry: Amazon web Services, Google app engine, Microsoft Azure.

PRACTICES:

- Performing hardware virtualization using Vmware workstation.
- Launch Amazon Linux EC2 Instance and connect the windows client to it.
- Launch Windows EC2 instance in AWS and connect windows client to it.
- Configure Web Server on Amazon Linux instance with Elastic IP.
- Manage Elastic Block Storage(EBS).
- Configure Amazon Simple Storage Service (Amazon s3).
- Configure Amazon S3 Glacier.
- Configure Amazon EFS.

MODULE-2

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

ANEKA

Cloud application platform, Framework overview, Anatomy of the Aneka container, Building Aneka clouds, Cloud programming, and management.

High Throughput Computing- Task Programming: Task computing, Task-based application models, Aneka task-based programming.

SKILLS:

- ✓ *Gain knowledge of different types of Cloud Service Providers.*
- ✓ *Explore basic design issues of Cloud Applications.*
- ✓ *Compare & evaluate the optimum costs in the data transmissions.*

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

CLOUD APPLICATIONS

Scientific applications in healthcare, biology, geo science; Business applications in CRM and ERP, productivity, social networking, media applications, multiplayer online gaming.

PRACTICES:

- Configure Amazon Virtual Private Cloud (VPC).
  a) Create your own VPC.
  b) Create a public subnet.
  c) Create a private subnet.
  d) Create an Internet gateway and attach to your VPC.
  e) Create Pubic Routing Table, associate subnet and add routing rules.
  f) Create Private Routing Table, associate subnet and add routing Rules.
  g) To launch Windows instance in Public subnet.
- Configure Amazon Elastic Load Balancer.
- Configure Relational Database Service (RDS).

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Deploying a VM Image to understand the evolution of cloud computing in contrast to the traditional approach.	Apply	1	1,5
2	Evaluate the concepts of various virtualization technologies.	Evaluate	1	2,5
3	Analyze the trade-offs, security, and privacy issues among application deployment in the various cloud and the local infrastructure.	Analyze	2	2,5
4	Deploy applications over commercial cloud computing infrastructures.	Apply	2	1,5

TEXT BOOKS:

1. Raj Kumar Buyya, C Vecchiola and S TSelvi , “Mastering Cloud Computing”, 1st Edition, Tata McGraw Hill Education (India), 2013.
2. RajKumarBuyya, Broberg J and GoscinskiA, “Cloud Computing - Principles and Paradigms”, 1st Edition, Wiley, 2011.

REFERENCE BOOKS:

1. David S. Linthicum, Cloud Computing and SOA Convergence in Your Enterprise A Step-by-Step Guide, Pearson 2010.
2. Dr. Kumar Saurabh, Cloud Computing, 2nd Edition, Wiley India 2012.
3. Rittinghouse J W, and Ransome J F, “Cloud Computing - Implementation, Management, and Security”, 1st Edition, CRC Press, 2009.
4. Michael Wittig and Andreas Wittig, “Amazon Web Services in Action”, 2nd Edition, Manning Publications, 2015.
5. Tim Mather, Subra Kumaraswamy, Shahed Latif, “Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance”, O’ReillyMediaInc, 2009.

DEPT. ELECTIVES

COMPUTER SCIENCE AND ENGINEERING

B.Tech.

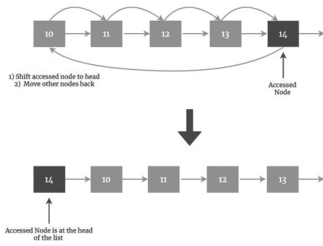
DEPARTMENT ELECTIVES

- 22CS801 - Advanced Data Structures
- 22CS802 - Advanced JAVA Programming
- 22CS803 - Computer Graphics
- 22CS804 - Deep Learning
- 22CS805 - Digital Forensics
- 22CS806 - Digital Image Processing
- 22CS807 - Web and Database Security
- 22CS808 - Machine Learning
- 22CS809 - Mobile Ad-hoc Networks
- 22CS810 - Mobile Application Development
- 22CS811 - Text Mining
- 22CS812 - Numerical Algorithms
- 22CS813 - Operating System Design
- 22CS814 - Optimization Techniques
- 22CS815 - Intrusion Detection and Prevention System
- 22CS816 - Simulation and Modeling
- 22CS817 - Parallel and Distributed Computing

COURSE CONTENTS

I SEM & II SEM

22CS801 ADVANCED DATA STRUCTURES



Source: https://www.baeldung.com/cs/advanced-data-structures

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Data structures.

COURSE DESCRIPTION AND OBJECTIVES:

Advanced data structures are one of the most important disciplines of data science since they are used for storing, organizing, and managing data and information to make it more efficient, easier to access, and modify. The objective of this course is to improve students' skills in designing data structures and algorithms for various solutions, and problem design in large systems and applications such as databases, information retrieval systems, bioinformatics, and geographic information systems.

MODULE-1

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

HASHING, EXTERNAL SORTING & PRIORITY QUEUE

Hashing: Introduction, Hash Function, Collision Resolution Technique: Separate Chaining, Open Addressing.

External Sorting: Merge Sort.

Priority: Priority Queues: Introduction, Types of priority queues, implementing max priority queue and min priority queue.

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

TREES

Trees: Splay tree, Red Black Tree (RBT), Operations on RBT, M-way search tree and operations, Segment tree.

PRACTICES:

- Implementation of Hash Table.
- Implementation of Separate Chaining and Linear Probing
- Implementation of merge sort by taking a large file as input and sort the file.
- Implement different operations on Priority Queue. i.e. adding an element, removing an element, size of the priority queue, printing the queue, and top element of the queue.
- Implementation of a Splay tree operation.
- Implementation of a Red-Black tree operation.
- Implementation of M Way Search tree.
- You are given a tree with n vertices (numbered 1, 2, …, n) and an integer k.

A subtree is defined as a connected subgraph of the tree. That is, a subtree is another tree that can be obtained by removing some (possibly none) vertices and all edges incident to those vertices from T.

A subset S of vertices is called good if every subtree containing all the nodes in S has at least k nodes.

Find the size of the smallest good subset.

Input:

The first line contains a single integer T, the number of test cases. The descriptions of test cases follow.

The first line of each test case contains two integers, n, and k.

The next n-1 lines each contain two integers u, v, denoting an edge between vertices u and v of the tree..

SKILLS:

- ✓ *Analyse the data structure required for various applications.*
- ✓ *Usage of trees, graphs, heaps and tries.*
- ✓ *Know various pattern searching algorithms.*

Output:

For each test case print in a single line, the minimum size of a good subset.

MODULE-2

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

HEAPS

Heaps: Introduction, types of heaps, heap implementation with an array, heap sort, technique to sort elements, Mergeable heaps, Application of heap.

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

PATTERN SEARCHING & TRIES

Pattern Searching: Pattern matching algorithms -Brute force, the Boyer —Moore algorithm, Robin-Karp algorithm, Knuth-Morris-Pratt algorithm.

Tries: Standard Tries, Compressed Tries, Suffix tries.

PRACTICES:

- You are given N numbers you can decrease each number by 10 % or K kg whichever is more. If the number, you choose is less than K you can make it zero. Your task is to minimize the sum of the number as small as possible by performing N operations only. [Note: Use Priority Queue].

  Example:

  Input:

  N = 2

  k = 10

  arr = 100 15

  Output:

  95

  Explanation:

  You will remove 10 from 100

  then once again remove 10 90.
- Given a Binary Heap of size N in an array arr[]. Write a program to calculate the height of the Heap.

  Input: N = 6

  arr = {1, 3, 6, 5, 9, 8}

  Output: 2
- Given two binary max heaps as arrays, merge the given heaps to form a new max heap.

  Example:

  Input :

  n = 4 m = 3

  a[] = {10, 5, 6, 2},

  b[] = {12, 7, 9}

  Output :

  {12, 10, 9, 2, 5, 7, 6}
- In Doraland, people have unique Identity Numbers called D-id. Doraemon owns the most popular gadget shop in Doraland. Since his gadgets are in high demand and he has only K gadgets left he has decided to sell his gadgets to his most frequent customers only. N customers visit his shop and D-id of each customer is given in an array array [ ]. In case two or more people have visited his shop the same number of time he gives priority to the customer with higher D-id. Find the D-ids of people he sells his K gadgets to.

Example:

Input:

N = 6

array[] = {1, 1, 1, 2, 2, 3}

K = 2

Output:

1 2

Explanation:

Customers with D-id 1 and 2 are most frequent.

- You are given q queries of two types:

  X: Append value X into an array.

  X K: You are required to print the Kth minimum XOR of X with the current array.

  You have to make a new array whose ith element is current_array[i]X. Then sort it and print the Kth element.

  Input format

  The first line contains q (1d”qd”100000).

  Next q lines contain the types of queries. (1 or 2) If type is 1, then it contains X (1d”Xd”10e18).

  If type is 2, then it contains X (1d”Xd”10e18) and K (Kd”current array size).

  Output format

  Print the number in the second type of query. (Note: Use Tries)

- Given an array of strings arr[] of size n and given s a string str and an integer k. The task is to find the count of strings in arr[] whose prefix of length k matches with the k length prefix of str. [Note: Use Tries]

  Input:

  n = 6

  arr[] = {“abba”, “abbb”, “abbc”, “abbd”, “abaa”, “abca”}

  str = “abbg”

  k = 3

  Output: 4

  Explanation:

  “abba”, “abbb”, “abbc” and “abbd” are the matching strings.

- Given an array of integers of size N find minimum xor of any 2 elements. [Note: Use Tries]

  Input:

  N = 3

  arr[] = {9,5,3}

  Output: 6

  Explanation:

  There are 3 pairs –

  9^5 = 12

  5^3 = 6

  9^3 = 10

  Therefore output is 6.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Implement external sorting and priority queue	Apply	1	1,2
2	Implement different types of trees and apply them to problem solutions.	Apply	1	1,2,3
3	Apply the concepts of advanced Trees for solving problems effectively.	Apply	1	1,2,3
4	Analyze how efficient pattern matching principles can be used in the design and implementation of qualitative research.	Analyse	2	1,2,3,4
5	Analyze the given scenario and choose the appropriate Data Structure for solving problems.	Analyse	2	1,2,3,4

TEXT BOOKS:

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson, 2004.
2. T.Cormen, R.Rivest, C.Stein, C.Leiserson,“Introduction to Algorithms”, , PHI publication, Second Edition, 2004, ISBN 81-203-2141-3.

REFERENCE BOOKS:

1. Mark de Berg, Otfried Cheong, Marc van Kreveld, Computational Geometry: Algorithms and Applications:, Springer; 3rd Edition. 2008.
2. Sahani, Anderson freed, Horowitz, Fundamentals of DATA STRUCTURES in C, Silicon Pr; 2nd Edition, August 1, 2007.

22CS802 ADVANCED JAVA PROGRAMMING

Source: https://www.kpl.gov/catalog/item/?i=ent://LYNDA/0/LYNDA:103608

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Programming JAVA.

COURSE DESCRIPTION AND OBJECTIVES:

Advanced Java programming covers the standard concepts such as database connectivity, web-services, web application development etc. It is specially designed to develop web-based, network-centric or enterprise applications. It simplifies the complexity of building an n-tier application.

MODULE-1

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

INTRODUCTION TO JAVA WEB FRAME WORKS

Introduction to Java Web frameworks: What is Framework in Java, Advantages of Frameworks, Popular Java Web Frameworks: Spring, Spring Boot, Hibernate, Java Server Faces, Google Web Toolkit etc., Comparison of the web frameworks. Environmental setup to use Web Frameworks.

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

ORM AND HIBERNATE

ORM: What is Object Relational Mapping, How ORM Works, Features of ORM, Advantages, Java ORM- Hibernate, JAVA Persistence API(JPA), ORM implementation.

Hibernate: Overview of Hibernate, Hibernate Architecture, Hibernate Mapping Types, Hibernate O/R Mapping, Hibernate Annotation, Hibernate Query Language.

PRACTICES:

- Installation and Environmental setup to work with java Web Frameworks.
- Create a simple application of hibernate using XML.
- Create a maven based hibernate application using annotation
- Develop a web application using hibernate framework.
- Using the web application to perform database access with hibernate framework.

MODULE-2

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

STRUTS

Struts: Struts framework, Struts features, Model 1vs Model 2 (MVC) Architecture, Core components of Struts, Configuration of Struts, Creating a Struts Application.

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

SPRING AND SPRING BOOT

Spring: Overview of Spring, Spring Architecture, Inversion of Control (IOC) and Dependency Injection, XML Configuration on Spring, Creating a Spring Application. Spring MVC, flow of Spring Web MVC, Spring Web MVC Framework Example.

Spring Boot: Overview of Spring Boot, Spring vs Spring MVC vs Spring Boot, SB architecture, Components of SB, Creating Spring Boot Application.

SKILLS:

- ✓ *To know the importance of java web frame-works to develop enterprise java applications.*
- ✓ *To experience developing a web application using MVC architecture.*
- ✓ *Design and develop CRUD applications using different java frameworks.*
- ✓ *To derive hands on experience of developing enterprise applications using advanced web frameworks like Spring and Spring Boot.*

PRACTICES:

- Develop a simple web application using struts.
- Develop a CRUD (Create Read Update Delete) Application using spring.
- Implement Spring MVC Pagination, which is used to display a large number of records in different parts. Display 10, 20 or 50 records in one page.
- Using Spring MVC upload a file.
- Create a self-contained HTTP application that uses embedded server like Tomcat using Spring Boot.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Implement web frameworks to develop web applications using java.	Apply	1	1,2,3
2	Design java web applications using MVC architecture using web frameworks like hibernate.	Create	1	1,2,3
3	Implement java application to interact with database using struts.	Apply	2	1,2,3
4	Design secure transaction-based web applications using Spring framework.	Create	2	1,2,3,4

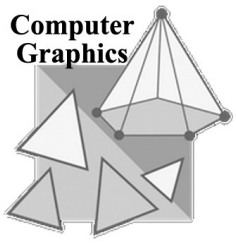
TEXT BOOKS:

1. James Keogh, “Complete Reference J2EE”, 9th Edition, McGraw Hill, Education, 2002.
2. Jeff Linwood and Dave Minter, “Beginning Hibernate”, 2nd Edition, Après publication 2010.

REFERENCES BOOKS:

1. Sharanam Shah, Vaishali Shah. “Structs 2 for Beginners”, 3rd Edition, Arizona Business Alliance, 2014.
2. Craig walls, “Spring in Action”, 5th Edition, Manning Publication, 2018.
3. https://struts.apache.org/getting-started/index.html.
4. https://www.javatpoint.com/hibernate-tutorial.
5. https://www.dineshonjava.com/spring-tutorial/.

22CS803 COMPUTER GRAPHICS



Source: https://www.javatpoint.com/computer-graphics-tutorial

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Basic Logical Thinking and Problem Solving Ability.

COURSE DESCRIPTION AND OBJECTIVES:

This course provides a comprehensive introduction to computer graphics leading to understanding the contemporary terminology and algorithms of computer graphics. To make the students learn the basic principles of visualization. To give an introduction to 2D and 3D modelling and animation.

MODULE-1

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

OVERVIEW OF GRAPHICS SYSTEM INTRODUCTION

Application areas of Computer Graphics; Video-display devices: raster-scan systems and random scan systems; Graphics primitives: display devices, primitive devices; Filled area primitives: scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.

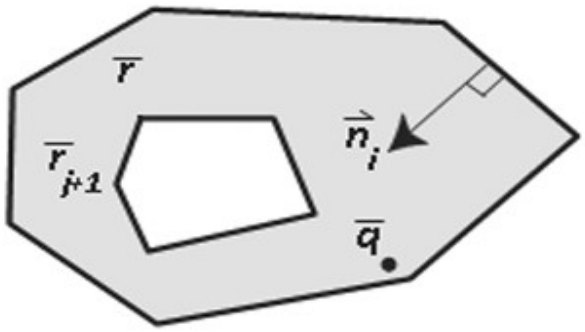
UNIT-2 **8L+8T+8P=24 Hours**

2D GEOMETRIC TRANSFORMATIONS AND VIEWING BASIC TRANSFORMATION

Geometric Transformations, matrix representations, and homogeneous coordinates. Inverse transformations, 2DComposite transformations, other 2D transformations, raster methods for geometric transformations; 2-D Viewing: The viewing pipeline, window to viewport coordinate transformation, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

PRACTICES:

- **Points & Polygons:** Suppose the vertices of a convex polygon are $\bar{p}1, \ldots, \bar{p}N$, given in counter-clockwise order. Let the coordinates of vertex $\bar{p}i$ be (xi ,yi).



Describe an algorithm that can tell whether a 2D point $\bar{q}$ is inside the gray-shaded region in the figure. You should assume that the polygons defined by vertices $\bar{p}1, \ldots, \bar{p}N$ and $\bar{r}1, \ldots, \bar{r}M$, respectively, are both convex. Hint: Each edge is contained in an infinite line. Each infinite line divides the 2D plane into two half-planes, the left half-plane and the right half-plane (here left/right means left/right with respect to a counter-clockwise direction of traversal of the vertices). The key insight you should use is that the interior of a convex polygon is the intersection of the left half-planes of all the edges of the polygon.

SKILLS:

- ✓ *Learn various clipping algorithms.*
- ✓ *Compare 2-D and 3_D transformations.*
- ✓ *Learn various surface detection methods*

- Transformations & Commutativity: We say that the 2D transformations f() and g() commute if and only if f(g($\bar{}$p)) = g(f($\bar{}$p)) for all points p$\bar{}$ ∈ R 2 . For each of the four cases below, where f() and g() are homographies, prove whether or not they commute:

  a) Both f() and g() are arbitrary homographies.

  b) One is an arbitrary rotation and the other an arbitrary translation.

  c) One is an arbitrary translation and the other is a non-uniform scaling.

  d) One is an arbitrary rotation and the other is a reflection. In each case, your solution can either be a derivation that proves/disproves commutativity or, if f() and g() do not commute, a specific counter-example.
- Contrast the implementation of Display File/Frame Buffer for a Random Scan System & Raster Scan System. Which type of system shall offer more consistent refresh rate and why?
- Determine the most appropriate pixels that will be plotted when Bresenham's algorithm is used to draw a line joining the points (10,20) and (20,30)
- Suppose a system with 8 inches by 10 inches video monitor that can display 100 pixels per inch. If memory is organized as one byte words, the starting frame buffer address is 0 and each pixel is assigned one byte of storage in memory, what is the frame buffer address of pixel with screen coordinates (x,y) ? Also, determine the total amount of memory consumed by the frame buffer.
- Consider a raster monitor of resolution 640*480 pixels. A scanning is used with horizontal retrace time of 4 micro seconds and vertical retrace time of 20 micro seconds respectively. Calculate the time available to display a pixel for both cases of (i) non-interlaced and (ii) interlaced. Assume a scan rate of 50 frames.

MODULE-2

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

3D GEOMETRIC TRANSFORMATIONS

3-D Object representation, 3-D geometry primitives, transformations, projection clipping. Interaction: Hardware input devices handling algorithms, Event handling echoing, Interactive techniques,3-D Geometric transformations, Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3D viewing: Viewing pipeline, viewing coordinates, general projection transformation and clipping.

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

VISIBLE SURFACE DETECTION METHODS

Back-face detection, depth buffer, A-buffer, Z-buffer, scan-line Illumination Models, and Surface rendering Methods: Basic illumination models, polygon rendering method.

PRACTICES:

- Matrix implementation is a powerful technique used for computer graphics. For the concerns stated in section II, matrix for 3D space is represented as a 4X4 array instead of 3X3 array.
- Each operation of object such as translation and rotation can be represented by one Matrix. By multiplying the matrix, a vector/point can be transformed by the operation. For example, P = (x, y, z, 1) is a point's coordinator before the operation, and M = (1 0 0 a) (0 1 0 b)(0 0 1 c) (0 0 0 1) is the matrix representation of translating the object a blocks right, b blocks up and c blocks forward, then after the translating, the point's coordinator will become P' = P*M is the coordinator of the point after the operation.
- A sequence of operation can also be composed to be one operation. For example, matrix M1 represent first operation, matrix M2 represent second operation, the matrix M = M1*M2 will represent this sequence of 2 operations. If P = (x, y, z, 1) is the coordinator of the point before perform the sequence of the operations, then P' = P * M will be the coordinator of the point after the perform the operation1 then perform the operation2.

Following are matrix representation for some operations:

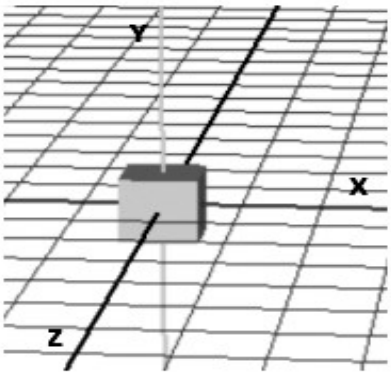
- Translate object by V, where V is the vector:

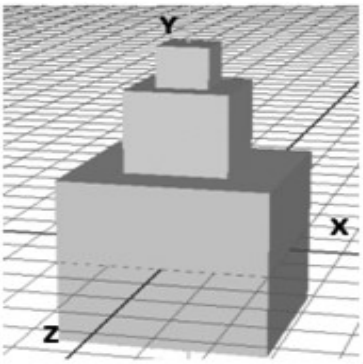
  M = (1 0 0 V[0]) (0 1 0 V[1]) (0 0 1 V[2]) (0 0 0 1)
- Rotate object along X axis by d degree:

  M = (1 0 0 0) (0 cos(d) -sin(d) 0) (0 sind(d) cos(d) 0) (0 0 0 1)
- Rotate object along Y axis by d degree:

  M = (cos(d) 0 sin(d) 0) (0 1 0 0) (-sin(d) 0 cos(d) 0) (0 0 0 1)
- Rotate object along Z axis by d degree:

  M = (cos(d) -sin(d) 0 0) (sin(d) cos(d) 0 0) (0 0 1 0) (0 0 0 1)
- Start with a unit cube centered at the origin. Use two more cubes like this one and obtain the model shown in the right image, where each new cube is twice the size of the previous one

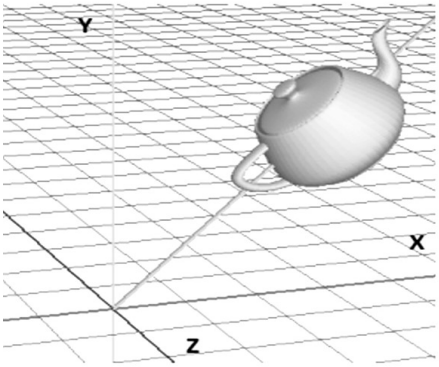


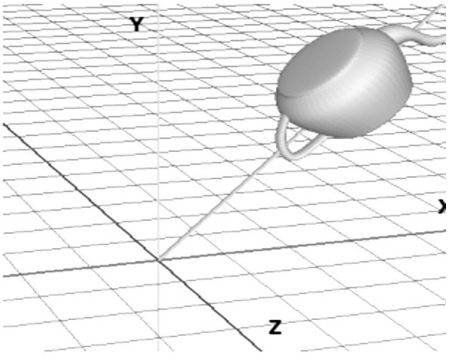


Below list the transformations in the order in which they were applied

1) scaling 2) Translation 3) Rotation

- There are situations in which rotations about axes other than the three coordinate axes are useful. In the next two activities, you will discover how a rotation about an arbitrary axis can be broken down into a series of rotations about the three coordinate axes.
- The yellow axis passing through the teapot in the image below is parallel to the vector (1, 1, 0). Determine a sequence of coordinate axis rotations that result in the teapot rotating 180 degrees about the yellow axis, as shown in the right image.





Write the sequence of rotations in the order in which you applied them. Hint: use a rotation that aligns the yellow axis with the x-axis, then rotate the object about the xaxis, then use the reverse rotation that brings the yellow axis back to original position.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Implement of various graphics systems along with the algorithms used for these devices for drawing.	Apply	1	1, 2
2	Demonstrate 2D graphics and algorithms including line drawing, polygon filling, clipping, and transformations.	Apply	1	1, 2
3	Demonstrate Concepts and techniques used in 3D computer graphics and basic about animation.	Apply	2	1, 2, 3
4	Implement Surface detection methods to get a realistic screen image.	Apply	2	1, 2,3

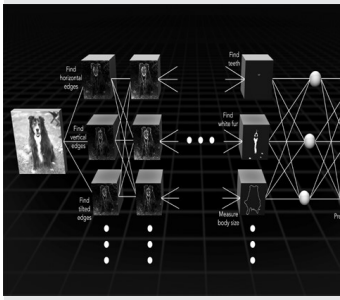
TEXT BOOKS:

1. Donald Hearn and M. Pauline Baker, “Computer Graphics C version”, Pearson education, Second Edition, 2014.
2. Zhigang Xiang, Roy Plastock, “Computer Graphics”, Schaum’s outlines, Second Edition, Tata McGraw Hill Edition, 2000.

REFERENCE BOOKS:

1. Foley, VanDam, Feiner and Hughes, “Computer Graphics Principles &Practice in C”, Pearson Education. 2nd Edition, 1996.
2. David F Rogers, “Procedural elements for Computer Graphics”, Tata McGraw Hill, 2nd Edition., 1988.
3. Neuman and Sproul, “Principles of Interactive Computer Graphics, Tata McGraw Hill”, 2nd Edition., 1978.
4. Shalini, Govil-Pai, “Principles of Computer Graphics”, Springer. 1st Edition, 2006.
5. Steven Harrington, “Computer Graphics”, TMH, 2nd Edition, 1987.

22CS804 DEEP LEARNING



Source: https://www.symmetrymagazine.org/sites/default/files/images/standard/neural_network_visual_final.jpg

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Machine learning, Python programming.

COURSE DESCRIPTION AND OBJECTIVES:

This course offers sufficient details required to understand the basic building blocks of various deep learning based models. Especially, focuses on different types of neural network models like feed forward neural networks, convolutional neural networks, recurrent neural networks, and deep auto encoders. During this course the students build, train, and evaluate deep neural network models for various applications in image, text, and speech domains. In addition, throughout this course students will be able to understand hyper parameter tuning and other best-practices to be followed while training deep neural network models.

MODULE-1

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

EVOLUTION OF DEEP NEURAL NETWORKS

Deep Learning Intuition: History of Deep Learning, what is Deep Learning? Applications of Deep Learning.

Neural Network Basics: McCulloch–Pitts neuron, Perceptron learning rule, Perceptron convergence theorem, Sigmoidal neuron, Multi-layer feed forward neural network, back propagation algorithm, Gradient descent method, Stochastic gradient descent method. Shallow Neural Networks and Deep Neural Networks.

REGULARIZATION and OPTIMIZATION for training Deep Models: Optimization methods - Adagrad, Adadelta, RMSProp, Adam; Regularization Methods-Dropout, Drop connect, Batch normalization; Activation functions - Linear, sigmoid, sigmoid, ReLU and variations of ReLU; Losss Function, Improving the training process – Dataset Augmentation, Noise Robustness, Weight Initialization methods, Early stopping, Parameter sharing and tying, bagging and other ensemble methods;

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

CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs): Foundations of Convolutional Neural Network, Popular Deep **CNN Models:** LeNet, AlexNet, VGGNet, ResNet, Google Net and other architectures.

INSTRUCTIONS FOR PRACTICES:

- Practice Assignments can be implemented using the Keras / Tensorflow APIs of Python
- Relevant data sets can be downloaded from standard repositories such as Kaggle/UCI or can be developed by the students.

PRACTICES:

- Implement Logistic regression With Neural Network Mindset.
  - logistic regression classifier for classification.
  - Plot the loss over each epoch.
  - Plot the accuracy over each epoch.
  - Report final Accuracy.

SKILLS:

- ✓ *Developing Vision and text based applications*
- ✓ *Hyperparameter Tuning of a deep Neural network model.*
- ✓ *Tensor Flow/ Keras tool usage for neural network implementation.*

- Implement Shallow Neural Network model:
  - Implement a binary classification neural network with a single and multiple hidden layers.
  - Implement a Multi-class classification neural network with a single and multiple hidden layers.
  - Vary the number of neurons at suitable layers.
- Hyper parameter Tuning of a Neural Network model implemented for hand-written digit classification:
  - Vary the type of activation functions.
  - Choose suitable Loss functions.
  - Vary the number of neurons at suitable layers.
  - Vary Weight Initialization methods.
  - Save the Best Model and load the saved model.
- Building a Deep Neural Network:
  - Implement a multi-class classification neural network with number of layers of your choice.
  - Include Batch Normalization layers.
  - Vary Optimization methods.
  - Add drop out layers.
- Convolutional Neural Network Models.
  - Design a Convolutional neural network with the layers of your choice
  - Compare the performance by changing the
  - Kernel size
  - Number of feature maps at each convolutional layer
  - Stride.
  - Padding.
  - Number of fully connected layers.
- Visualization of CNN Models.
  - Design a Convolutional Neural Network Model for image classification.
  - Plot Model Architecture.
  - Visualize feature maps after training of CNN.
  - Visualize class activation maps.

MODULE-2

UNIT-1 **14L+0T+10P=24 Hours**

DEEP UNSUPERVISED LEARNING

Transfer learning Approaches: Deep Pre-trained architectures- AlexNet, VGG16, VGG19, ResNEt. Use deep Convolutional architectures for feature extraction and fine-tuning tasks.

Deep Unsupervised Learning: Autoencoders- Under complete Autoencoders, regularized auto encoders, Representation power, layer size and depth, stochastic encoders and decoders, Denoising auto-encoders, Sparse auto encoder, Contractive auto-encoders

UNIT-2 **10L+0T+6P=16 Hours**

RECURRENT NEURAL NETWORKS

Architecture of an RNN, unfolding of an RNN, Backpropagation through time, Long short term memory (LSTM), Gated recurrent units, Applications- Text Classification, Sentiment Analysis.

PRACTICES:

- Using Deep pre-trained CNN model for feature extraction:
  - Extract features from the FC1 of VGG network.
  - Train any traditional ML model like SVM for classification.
  - Repeat the above by considering FC2 of VGG for feature extraction.

- Fine-tuning Deep pre-trained CNN for Classification:
  - Fine-tune VGG network for the task under consideration.
  - Check the performance by making.
  - all the layers trainable.
  - freezing the initial layers.
  - freezing the entire network except the final layer.
- Design MLFFNN with 3-level stacked autoencoder based pre-training for Black and white image data, Display features extracted by different levels of stacked autoencoder at the end of pre-training.
- Sentiment Analysis
  - Pre-process the text.
  - Convert the text into word embeddings.
  - Implement the classification network using LSTMs/ GRUs.
  - Pre-process the text.
  - Convert the text into word embeddings.
  - Implement the classification network using LSTMs/ GRUs.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Implementation of Deep learning models to solve various real-time problems.	Apply	1,2	1, 3
2	Analyse performance of a deep network and tune its capacity and hyper parameters.	Analyze	1	2
3	Leveraging tools to Build deep networks and apply them for real word tasks.	Apply	1,2	1,5
4	Developing core components for deep learning algorithms.	Design	1,2	3

TEXT BOOKS:

1. Ian Good Fellow and Yoshua Bengio and Aaron, “Deep Learning”, 1st Edition, MIT Press, 2016.
2. Charu C Aggarwal “Neural Networks and Deep learning”, Springer International Publishing, 2018.

REFERENCE BOOKS:

1. Francois Chollet, “Deep learning with python”, 1st edition, Manning Publications, 2017.
2. S. Haykin, “Neural Networks and Learning Machines”, 3rd edition, Prentice Hall of India, 2011.
3. Josh Patterson and Adam Gibson, “Deep Learning: A Practitioner’s Approach”, 1st Edition, O’Reilly, 2017.
4. Satish Kumar, “Neural Networks, A Classroom Approach”, Tata McGraw-Hill, 2007.

Source: https://www.electrosoft-inc.com/resources/digital-forensics

22CS805 DIGITAL FORENSICS

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Cyber Security and Cyber Laws, Cryptography and Network Security.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on understanding forensic terminologies and approaches along with variety of tools used for digital forensic investigations. The objective of this course is to understand digital forensics and its usage in solving computer crimes. By end of the course, students will be able to identify improper usage of computer systems and legal concepts in digital forensic investigation stages.

MODULE-1

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

INTRODUCTION

Computer forensics fundamentals, Benefits of forensics, computer crimes, computer forensics evidence and courts, legal concerns and private issues, Steps of computer forensics.

Understanding Computing Investigations: Procedure for corporate High-Tech investigations, understanding data recovery workstation and software, conducting and investigations.

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

DATA ACQUISITION

Understanding storage formats and digital evidence, determining the best acquisition method, acquisition tools, validating data acquisitions, performing RAID data acquisitions, remote network acquisition tools, other forensics acquisitions tools.

PRACTICES:

- Recover Deleted Files using Forensics Tools.
- Study the steps for hiding and extract any text file behind an image file/ Audio file using Command Prompt.
- Extract Exchangeable image file format (EXIF) Data from Image Files using Exifreader
- Software How to make the forensic image of the hard drive using EnCase Forensics.
- Restoring the Evidence Image using EnCase Forensics.

MODULE-2

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

PROCESSING CRIMES AND INCIDENT SCENES

Securing a computer incident or crime, seizing digital evidence at scene, storing digital evidence, obtaining digital hash, reviewing case.

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

CURRENT COMPUTER FORENSICS TOOLS

Software, hardware tools, validating and testing forensic software, addressing data-hiding techniques, performing remote acquisitions, E-Mail investigations- investigating email crime and violations, understanding E-Mail servers, specialized E-Mail forensics tool.

PRACTICES:

- Study of Computer Forensics and different tools used for forensic investigation.
- Live Forensics Case Investigation using Autopsy.
- Collect Email Evidence in Victim PC.
- Extracting Browser Artifacts.
- View Last Activity of Your PC.
- Find Last Connected USB on your system (USB Forensics).
- Comparison of two Files for forensics investigation by Compare IT software.

SKILLS:

- ✓ *Identify sources and methods of system intrusion.*
- ✓ *Create bitwise images of hard drives for forensic analysis.*
- ✓ *Collecting evidences from log files.*

COURSE OUTCOMES:

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

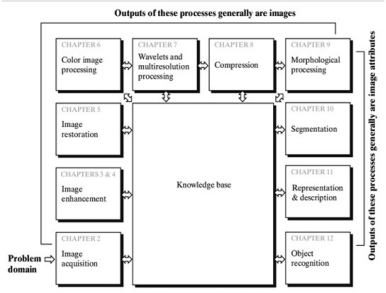
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply basic legal concepts related to digital forensics and evidence collection.	Apply	1	1,2,3
2	Analyze various digital forensics frameworks and its usage to solve crimes.	Analyse	1	1,2,3
3	Analyze artifacts like logs, packet captures, and registry.	Analyse	2	1,2,3
4	Demonstrate the ability to use forensic tools.	Create	2	1,2,5
5	Design and develop various forensic applications using variety of tools to carryout forensic investigation.	Analyse	2	1,2,3

TEXT BOOKS:

1. Warren G. Kruse II and Jay G. Heiser, “Computer Forensics: Incident Response Essentials”,Addison Wesley, 2020.
2. Nelson, B, Phillips, A, Enfinger, F, Stuart, C., “Guide to Computer Forensics and Investigations, 6th edition, Thomson Course Technology, 2020, ISBN: 0-619-21706-5.

REFERENCE BOOKS:

1. Vacca, J, Computer Forensics: Computer Crime Scene Investigation, 2nd Ed, Charles River Media, 2015, ISBN: 1-58450-389.



Source: https://www.researchgate.net/figure/Fundamental-steps-in-digital-image-processing_fig10_333856607

22CS806 DIGITAL IMAGE PROCESSING

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Probability and Statistics.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on imparting knowledge about the aspects of Image Processing and its applications. The main objective of the course is to learn digital image fundamentals, image transforms, image enhancement, restoration and compression, morphological image processing, representation and description.

MODULE-1

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

FUNDAMENTALS OF IMAGE PROCESSING

Fundamental steps in digital image processing, Components of image processing system, A simple image formation model, Image sampling and quantization, Basic relationships between pixels, Introduction to Fourier Transform and DFT – properties of 2D Fourier Transform, FFT.

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

IMAGE ENHANCEMENT IN THE SPATIAL AND FREQUENCY DOMAINS

Basic gray - level transformations, Histogram processing, Basics of spatial filtering, Smoothing spatial filters, Sharpening spatial filters, The basics of filtering in the frequency domain, Image smoothing in frequency domain filters, Image sharpening in frequency domain filters.

Image Segmentation: Fundamentals, Point, Line and edge detection, Thresholding, Region-based segmentation, Segmentation using morphological watersheds, The use of motion in segmentation.

PRACTICES:

- Develop a module to enhance the image by using image arithmetic and logical operations.
- Develop a module for an image enhancement using kernel operations.
- Develop a module for gray level slicing with and without background.
- Develop a module for image enhancement using histogram equalization.
- Develop a module to filter an image using low pass & high pass filter in spatial domain. Compare the performance of both filters.
- Develop a module for smooth an image using low pass & high pass filters in frequency domain. Compare the performance of both filters.
- Develop a module for detecting lines & edges in an image.
- Develop a module for segmenting region of interest.

MODULE-2

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

IIMAGE RESTORATION

A model of image degradation/restoration, Noise models, inverse filtering, wiener filtering, Constrained Least Squares Filtering, Geometric Mean Filter.

Image Compression: Fundamentals, Huffman coding, Golomb coding, LZW coding, Run-length coding

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

SKILLS:

- ✓ *Apply knowledge of science and engineering principles to image related problems.*
- ✓ *Undertake image problem identification and formulate solutions.*
- ✓ *Implement algorithms for enhancement, restoration, compression etc.*

MORPHOLOGICAL IMAGE PROCESSING

Erosion, Dilation, Opening, Closing, The hit-or-miss transformation; Basic morphological algorithms - boundary extraction, hole filling, extraction of connected components, thinning, thickening, skeletons, pruning.

Feature Extraction: Background, Boundary preprocessing, Boundary Feature Descriptors, Region Feature Descriptors, Principle Components as feature descriptors, Whole-image features.

PRACTICES:

- Develop a module to perform add & removal of salt and pepper noise. Compute PSNR & MSE and check the impact before and after removal of noise.
- Develop a module to remove noise using average filter and median filter. Compute PSNR & MSE before and after removal of noise.
- Develop a module for image compression and decompression.
- Develop a module for morphological image operations - erosion, dilation, opening & closing.
- Develop a module for morphological image operations - hit-or-miss transformation.
- Develop a module for morphological image operations - thinning, thickening
- Develop a module for extracting boundary features of an image.
- Develop a module for extracting features of an image using GLCM.

COURSE OUTCOMES:

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

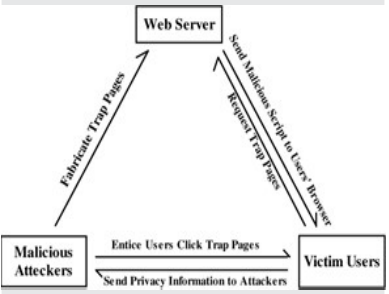
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Understand the fundamental concepts of a digital image processing system.	Under-stand	1	1,2
2	Learn different techniques employed for the en-hancement of images.	Analyse	1	1,2,3,5,12
3	Employ image segmentation and representation techniques to extract region of interest.	Apply	1	1,2,3,5,12
4	Learn different causes for image degradation and overview of image restoration techniques.	Evaluate	2	1,2,3,5,12
5	Apply various compression techniques to reduce image size and morphological operations to extract features.	Apply	2	1,2,3,5,12
6	Learn different feature extraction techniques for image analysis and recognition.	Apply	2	1,2,3,5,12

TEXT BOOKS:

1. Rafeal C Gonzalez and Richard E.Woods, “Digital Image Processing”, 4th edition, Pearson Education/ PHI, 2018.
2. Rafeal C Gonzalez and Richard E.Woods, “Digital Image Processing using MATLAB”, 4th edition, Pearson Education/ PHI, 2020.

REFERENCE BOOKS:

1. Milan Sonka, Vaclav Hlavac and Roger Boyle, “Image Processing, Analysis and Machine Vision”, 4th Edition, Cengage, 2015.
2. Alasdair McAndrew, “Introduction to Digital Image Processing with Matlab”, Thomson Course Technology, 2004 Course Technology Press, Boston, MA, United States, 2004.
3. William K. Prat, “Digital Image Processing”, 4th Edition, Wiley-Interscience, A John Wiley & Sons, Inc., Publication, 2007.



Source: https://www.intechopen.com/chapters/37306

22CS807 WEB AND DATABASE SECURITY

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Cryptography and Network Security, Web Technologies and Database Systems.

COURSE DESCRIPTION AND OBJECTIVES:

This course is imperative for understanding the fundamental security principles of the web and Data base security. The course provides an overview of the most common attacks, and illustrates fundamental countermeasures that every web application should implement. In essence, this course offers you the knowledge and skills to build better and more secure applications and to identify risks and vulnerabilities in operating systems from a database perspective.

MODULE-1

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

WEB SECURITY FUNDAMENTALS

Introduction to the web security landscape, and an overview of the most relevant threats. Understanding the security model of the web, and the recent evolution towards client-centric security.

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

SECURING THE COMMUNICATION CHANNEL

Understanding the dangers of an insecure communication channel. Practical advice on deploying HTTPS, and dealing with the impact on your application. Insights into the latest evolutions for HTTPS deployments.

Preventing Unauthorized Access: Understanding the interplay between authentication, authorization and session management. Practical ways to secure the authentication process, prevent authorization bypasses and harden session management mechanisms.

PRACTICES:

- Explore various tools for DOS attacks. (For Ex: SSPing, Land Exploit)
- Explore various tools for DDOS attacks. (For Ex: Trinoo, TFN)
- Identify various computer based social engineering ways to acquire sensitive information or inappropriate access privileges by an outsider
- Set up IPSEC under LINUX
- Implement a code to simulate buffer overflow attack.

MODULE-2

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

SECURELY HANDLING UNTRUSTED DATA

Investigation of injection attacks over time, understanding the cause behind both server-side and client-side injection attacks. Execution of common injection attacks, and implementation of various defenses.

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

ADMINISTRATION OF USERS

Introduction-Authentication-Creating Users, SQL Server User-Removing, Modifying Users-Default, Remote Users-Database Links-Linked Servers-Remote Servers-Practices for Administrators and Managers-Best Practices Profiles, Password Policies.

Privileges and Roles: Introduction-Defining and Using Profiles-Designing and Implementing Password Policies-Granting and Revoking User Privileges-Creating, Assigning and Revoking User Roles-Best Practice.

Database Application Security Models: Introduction-Types of Users-Security Models-Application Types-Application Security Models-Data Encryption.

PRACTICES:

- Explore Web Application Vulnerabilities.
- Implement Web based Password Cracking Techniques.
- Implement SQL Injection attack.
- Implement XSS attack.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze the cause behind both server-side and client-side injection attacks.	Analyze	1	1,2,,3
2	Analyze the privileges and roles of the users.	Analyze	2	1,2,3
3	Analyze common attacks and countermeasures.	Analyze	2	1,2,3
4	Implement administration policies for users, database security models.	Evaluate	2	1,2,3

TEXT BOOKS:

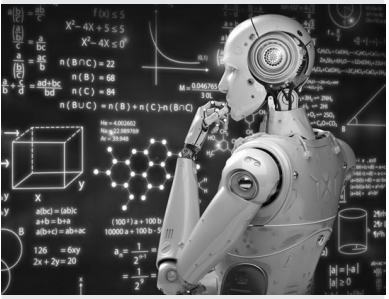
1. Bing Liu, “Web Data Mining”, 2nd edition, Spinger, 2011.
2. Hassan A. Afyouni, “Database Security and Auditing”, Third Edition, Cengage Learning, 2013.

REFERENCE BOOKS:

1. Jiawei Han, Micheline Kamber and Jian Pei, “Data Mining: Concepts and Techniques”, 3rdedition, Morgan Kaufmann Publishers, 2011.
2. Bing Liu, “Sentiment analysis and opinion mining”, 2nd edition, Morgan & Claypool Publishers, 2012.
3. Jure Leskovec, Anand Raja Raman and Jeffrey D Ullman, “Mining of Massive Datasets”, 5thedition, Stanford University, 2014.Networks, Springer.
4. Frank Adelstein, Sandeep K.S. Gupta, Golden G. Richard III, and Loren Schwiebert. Fundamentals of Mobile and Pervasive Computing, McGraw-Hill Professional, 2015.

SKILLS:

- ✓ *Investigation of injection attacks over time.*
- ✓ *Execution of common injection attacks and their defenses.*
- ✓ *Implement privacy preserving data mining algorithms.*



Source: https://www.forbes.com/sites/kalevleetaru/2019/01/15/why-machine-learning-needs-semantics-not-just-statistics/

22CS808 MACHINE LEARNING

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Probability and Linear algebra, Python programming.

COURSE DESCRIPTION AND OBJECTIVES:

This course provides a broad introduction to various machine learning concepts including Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks) and Unsupervised learning (clustering, dimensionality reduction) methods. Students will get an understanding of various challenges of Machine Learning and will be able to decide on model complexity. Numerous case studies introduced in this course allow the students to apply machine-learning algorithms in computer vision, medical imaging, audio, and text domains. Laboratory experiments of this course will introduce students to advanced Machine Learning Python libraries such as Scikit-Learn, Matplotlib, and many other recent ML-related APIs. The course is designed such that the students get enough hands-on experience with a major focus on the practical implementation of theoretical concepts.

MODULE-1

UNIT-1 **14L+0T+8P=22 Hours**

INTRODUCTION

What is machine learning? Machine learning applications; Types of Learning: Supervised learning; Un-supervised learning; Reinforcement learning.

Model Training Essentials: Re-sampling methods: Bias–Variance Trade-off. Hypothesis Testing and Variable Selection, Sub sampling and Upsampling, SMOTE; Cross Validation (validation set, Leave-One-Cut (LOO), k-fold strategies) and bootstrap; Evaluation measures-Error functions, Confusion Matrix, Accuracy, Precision and Recall, F1 Score.

Regression Analysis: Linear Regression, Simple and Multiple Linear Regression, Polynomial Regression, Logistic Regression, Multi nominal Regression. Ordinary Least Squares Method, Model Shrinkage-Ridge, and LASSO regression.

UNIT-2 **10L+0T+8P=18 Hours**

FEATURE SELECTION

Feature Selection Strategies: Problem statement and Uses, Filter methods, Wrapper methods, Embedded methods. Branch and bound algorithm, Sequential forward/backward selection algorithms.

Dimensionality Reduction: Singular value decomposition, matrix factorization, Linear discriminant analysis, Principal components analysis.

PRACTICES:

- Apply the following tasks to any given dataset:
  a) Load and visualize data.
  b) Check out and replace missing values.
  c) Encode the Categorical data.
  d) Splitting the dataset into Training and Test set.
  e) Splitting the dataset into k-folds.
  f) Feature scaling.

- House price prediction:
  a) Create a model that predicts a continuous value (price) from input features (square footage, number of bedrooms and bathrooms).
  b) Implement a univariate Model using Least Squares and plot best-fit line.
  c) Implement a multivariate Model using Least Squares and plot best-fit line.
  d) Retrieve model error and model coefficients.
  e) Observe Variance Inflation Factor (VIF).
  f) Implement Ridge regression model.
  g) Implement LASSO regression model.
  h) Report your observations on the above models for house prediction.
- Heart disease prediction:
  a) Implement a logistic regression model to predict whether an individual is suffering from heart disease or not.
  b) Evaluate and compare model performance using the following validation approaches:
  i. Validation set approach.
  ii. K-fold cross validation.
  iii. Stratified K-fold cross validation.
  iv. LOO strategy.
  c) Plot Confusion matrix.
  d) Report performance of the model in terms of the following metrics:
  i. Accuracy.
  ii. Precision-Recall.
  iii. F1 Score.
  e) Report your observations and explain when to use what type of measures.
- Implement the Polynomial Regression algorithm to fit data points. Select the appropriate data set for your experiment and draw graphs.
- Working with imbalanced datasets:
  a) Load an imbalanced dataset and visualize imbalance in the data as a bar plot.
  b) Implement KNN model for classification.
  c) Balance the dataset using:
  i. Random Over sampling.
  ii. Random Under sampling.
  iii. SMOTE.
  d) Implement KNN model for classifying data balanced in the above steps.
  e) Report your observations on the performance of models trained using balanced and imbalanced data.
- Perform effective feature selection in a given dataset using any one of the feature selection techniques.
- Dimension Reduction:
  a) Load a dataset and Implement Bayes classification model.
  b) Apply dimension reduction using:
  i. Principal Component Analysis
  ii. Linear Discriminant Analysis
  c) Apply the model on data with reduced dimension.
  d) Compare and contrast model performance in each case.

SKILLS:

- ✓ *Statistical data analysis.*
- ✓ *Classify / Cluster data.*
- ✓ *Tool usage for developing ML applications.*

MODULE-2

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

CLASSIFICATION

Classification: Binary, Multi-class and Multi-label Classification; K-Nearest Neighbours, Support Vector Machines, Decision Trees, The Naïve Bayes' Classifier, Class Imbalance, Perceptron ANN model.

Ensemble Methods: Ensemble Learning Model Combination Schemes, Voting, Error-Correcting Output Codes, Bagging: Random Forest Trees, Boosting: Adaboost, Stacking.

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

CLUSTERING

Clustering: Different distance functions and similarity measures, K-means clustering, Medoids, Hierarchical Clustering-Single linkage and Complete linkage clustering, Graph based Clustering -MST, DBSCAN, Spectral clustering.

PRACTICES:

- Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
- Implement the naïve Bayesian classifier for a sample training data set stored as a.csv file. Compute the accuracy of the classifier, considering few test data sets.
- Assuming a set of spam or not-spam mails that need to be classified, use the naïve Bayesian classifier model to perform this task. Calculate the accuracy, precision, and recall for your data set.
- Implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
- Demonstrate the working of the decision tree-based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample?
- Build a model using SVM with different kernels.
- Implement and build models using the following Ensemble techniques.
  a) Bagging.
  b) Boosting: Adaboost, Stacking.
- Build a model to perform Clustering using K-means after applying PCA and determining the value of K using the Elbow method.
- Unsupervised Modeling:
  a) Cluster the data using the following models:
  i. Spectral Clustering.
  ii. K-medoids.
  iii. DBSCAN.
  iv. Hierarchical Clustering.
  b) Compare and contrast model performance in each case.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply a wide variety of learning algorithms such as Probabilistic, Discriminative and Generative algorithms for a given application.	Apply	1, 2	1
2	Design an end-to-end Machine-learning model to realize solutions for real-world problems.	Apply	1	3
3	Implement various machine learning models using advanced ML tools.	Create	1, 2	5
4	Analyze and evaluate the performance of various machine learning models approaches on different kinds of data.	Analyze	2	2

TEXT BOOKS:

1. Ethem Alpaydin, “Introduction to Machine Learning”, 3rd Edition, The MIT Press, 2014.
2. Flach, Peter. “Machine learning: the art and science of algorithms that make sense of data”. Cambridge University Press, 2012.

REFERENCE BOOKS:

1. Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.
2. Aurélien Géron, “Hands-on Machine Learning with Scikit Learn and Tensor Flow”, O’reilly, 2017.
3. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, “An Introduction to Statistical Learning with Applications in R”, Springer, 2013. (ISLR).

Source: https://www.educba.com/mobile-ad-hoc-network/

22CS809 MOBILE AD-HOC NETWORKS

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Computer networks.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on major aspects of ad hoc networking, from design through performance issues to application requirements. It starts with the design issues and challenges associated with implementations of ad hoc network applications. This includes mobility, disconnections, and battery power consumption. The course provides a detailed treatment of proactive, reactive, and hybrid routing protocols in mobile wireless networks. It also covers the IEEE 802.11 Wireless LAN and discusses their characteristics and operations. Through activities, the course gives students hands-on experience in designing a mobile ad hoc network using the NS2 simulator.

MODULE-1

UNIT-1 **8L+0T+6P=14 Hours**

INTRODUCTION

Introduction to ad-hoc networks-definition, characteristics, features, applications; Characteristics of the wireless channel; Ad-hoc mobility models-indoor and outdoor models.

UNIT-2 **16L+0T+10P=26 Hours**

MEDIUM ACCESS PROTOCOLS

MAC protocols- design issues, goals and classification; Contention-based protocols – with reservation, without reservation; Scheduling algorithms; Protocols using directional antennas; IEEE standards - 802.11a, 802.11b, 802.11g, 802.15; HIPERLAN.

Network Protocols: Routing protocols - design issues, goals, and classification; Proactive Vs reactive routing; Unicast routing algorithms; Multicast routing algorithms; Hybrid routing algorithm; Energy-aware routing algorithm; Hierarchical routing; QoS aware routing.

PRACTICES:

- Installation of NS-2 and basics of TCL scripting.
- TCL script for
  a) computing the arithmetic operations on two operands.
  b) finding the given number is prime or not using functions.
  c) finding the factorial value of a given number.
- Set the node property and routing protocol in the same MANET scenario.
- Analyse the performance of the MANET.
- Develop MAC Protocol using any suitable Network Simulator for MANETs to send the packet without any contention through wireless link using the following MAC protocols (CSMA/CA (802.11)). Analyze its performance with increasing node density and mobility.
- Simulate MANET environment using suitable Network Simulator and test with various mobility model such as Random walk, Random waypoint and Group mobility. Analyze throughput, PDR and delay with respect to different mobility models.

MODULE-2

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

END-END DELIVERY AND SECURITY

Transport layer - issues in designing, transport layer classification, ad-hoc transport protocols; Security issues in ad-hoc networks - issues and challenges, network security attacks; Secure routing protocols.

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

CROSS LAYER DESIGN

Cross layer design - need for cross layer design, cross layer optimization; Parameter optimization techniques; Cross layer cautionary perspective; Integration of adhoc with mobile IP networks.

PRACTICES:

- Create CBR traffic over UDP and TCP.
- Write an awk script that takes data from trace file and give the report for performance metrics such as packet delivery ratio, and throughput.
- Implement Transport Control Protocol in Sensor Network.
- Design and Implementation of Security algorithm for Wireless networks (b)Implementation of security protocol for mobile network.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Summarize the protocols used at the MAC layer and scheduling mechanisms to express the mathematical properties.	Evaluate	1	1, 12
2	Apply proactive and reactive routing algorithms to find optimal paths.	Apply	1	1, 2, 5, 12
3	Analyze types of routing protocols used for unicast and multicast routing.	Analyse	1	1, 2, 5, 12
4	Compare the performance of various routing protocols in ad-hoc networks.	Analyse	2	1, 2
5	Develop the network security solution and routing mechanism.	Apply	2	1, 2, 12

TEXT BOOKS:

1. C. Siva Ram Murthy and B. S. Manoj, “Ad hoc Wireless Networks Architecture and Protocols”, 2nd Edition, Pearson Edition, 2007.
2. Charles E. Perkins, “Ad hoc Networking”, 1st Edition, Addison – Wesley, 2000.

REFERENCE BOOKS:

1. Stefano Basagni, Marco Conti, Silvia Giordano and Ivan stojmenovic, “Mobile ad-hoc networking”, Wiley-IEEE press, 1st Edition, 2004.
2. Mohammad Ilyas, “The Handbook of Adhoc Wireless Networks”, 1st Edition, CRC press, 2002.
3. T. Camp, J. Boleng and V. Davies “A Survey of Mobility Models for Ad Hoc Network Research” Wireless Commun. and Mobile Comp., Special Issue on Mobile Ad Hoc Net working Research, Trends and Applications, vol. 2, no. 5, 2002, pp. 483–502.
4. A survey of integrating IP mobility protocols and Mobile Ad hoc networks, Fekri M. Abduljalil and Shrikant K. Bodhe, IEEE communication Survey and tutorials, v no.1, 2007.
5. V. T. Raisinhani and S.Iyer “Cross Layer Design Optimization in Wireless Protocol Stacks” Comp. Communication, Vol 27 no. 8, 2004.

SKILLS:

- ✓ *Evaluate various routing protocols.*
- ✓ *Analyse the performance of MAC protocols for Ad-hoc networks.*
- ✓ *Analyse the performance of Network protocols for Ad-hoc networks.*

Source: https://www.tatvasoft.com.au/blog/mobile-application-development-methodology/

22CS810 MOBILE APPLICATION DEVELOPMENT

Hours Per Week :

L	T	P	C
2	0	4	4

PREREQUISITE KNOWLEDGE: OOPs through Java, DDL & DML Commands – DBMS..

COURSE DESCRIPTION AND OBJECTIVES:

This course guides the student in designing and building a mobile application using Android™. The main objective of this course is to let the student learn basic Android programming concepts while building a variety of apps, starting with basic to making use of advanced concepts.

MODULE–1

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

INTRODUCTION

Introduction to Mobile Application Development- Mobile Applications and Device Platforms, Alternatives for Building Mobile Apps; Introduction to Android, Android versions, Android Architecture.

Application Development Process- Developers Workflow basics, Installing the Android SDK Tools; Anatomy of an Android Application.

Basic Building blocks - Activities, Services, Broadcast Receivers & Content providers; Intents & Fragments.

View Group- Layout: Linear Layout, Relative Layout, Frame Layout, Grid Layout, constraint Layout, Table Layout, and Absolute Layout.

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

VIEWS

Views: Basic Views; Picker Views- Time Picker View, Data Picker View; List Views – List View, Spinner View; Scroll View.

Activities: Creating an activity, Understanding the activity life cycle using Log and Toast, applying styles and themes to an activity, and hiding the activity title.

Linking Activities using Intents: Introduction to Intents and its types with examples, passing data between activities with intents, Activity Navigation- Implement up navigation with parent activities.

Fragments: Introduction to Fragment, the life cycle of a fragment, Adding fragments dynamically, Interaction between fragments.

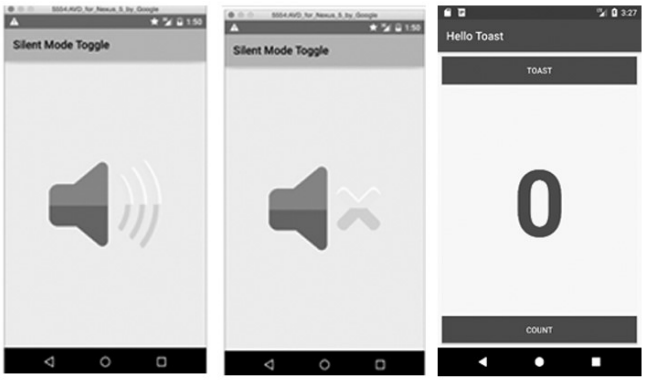
PRACTICES:

- Setting up Android Studio:
  a) Installing Android Studio
  b) Select an empty activity to simulate the “Welcome App” Using Android Studio.
  c) Exploring the interface of the Android Studio to understand the Project Structure.
- Develop an Android application using controls like Button, TextView, and EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.
- **Design the HelloToast app:** The HelloToast app consists of two Button elements and one Text View. When the user taps the first Button, it displays a short message (a Toast) on the screen. Tapping the second Button increases a “click” counter displayed in the TextView, which starts at zero.

SKILLS:

- ✓ *Design mobile applications for user requirements.*
- ✓ *Use of suitable advanced components to design mobile apps.*
- ✓ *Utilization of activities, intents, layouts, and views for content.*

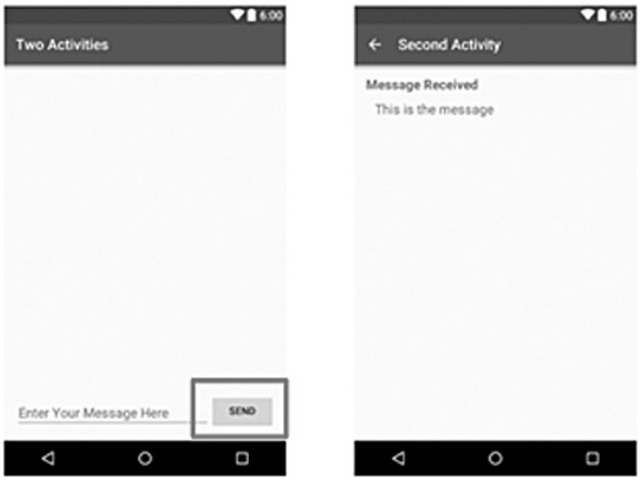
- **Design Silent Model Toggle application:** This app allows the user to toggle the ringer mode on the phone by simply pressing a button.



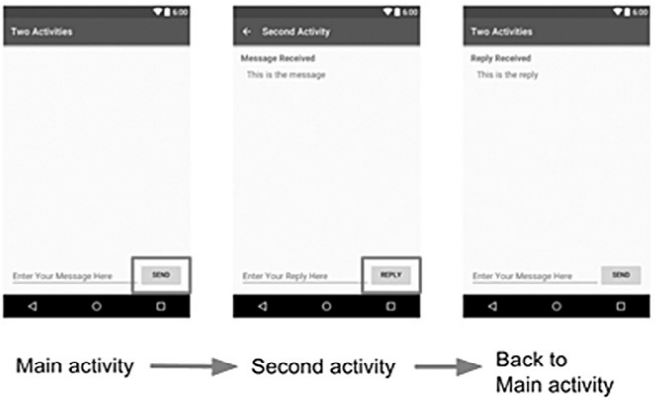
- In this assignment, students will create and build an app called Two Activities. Students will build the app in four stages.

a) In the first stage, you create an app whose main activity contains one button, Send. When the user clicks this button, your main activity uses an intent to start the second activity.

b) In the second stage, you add an EditText view to the main activity. The user enters a message and clicks Send. The main activity uses an intent to start the second activity and sends the user's message to the second activity. The second activity displays the message received.



c) In the final stage of creating the Two Activities app, you add an EditText and a Reply button to the second activity. The user can now type a reply message and tap Reply, and the reply is displayed on the main activity. At this point, you use an intent to pass the reply from the second activity to the main activity.



d) Implement all the Activity lifecycle call back methods to print messages to logcat when those methods are invoked. These log messages will allow you to see when the Activity lifecycle changes state, and how those lifecycle state changes affect your app as it runs.

- **Design an application with implicit intents:** Create a new app with one Activity and three options for actions: open a website, open a location on a map, and share a snippet of text. All the text fields are editable (EditText) but contain default values.

- **Design Droid Café:** In this practical, the student will create and build a new app starting with the Basic Activity template that imitates a dessert-ordering app. The user can tap an image to perform an action-in this case, display a Toast message-as shown in the figure below. The user can also tap a shopping cart button to proceed to the next Activity.
- **Experiment with the android:** inputType attribute for EditText elements. You add EditText elements for a person's name and address and use attributes to define single-line and multiple-line elements that make suggestions as you enter text. You also add an EditText that shows a numeric keypad for entering a phone number.
- Other types of input controls include interactive elements that provide user choices. You add radio buttons to Droid Cafe for choosing only one delivery option from several options. You also offer a spinner input control for selecting the label (Home, Work, Other, Custom) for the phone number.

MODULE–2

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

CREATING A FEATURE-RICH APPLICATION

Creating a Feature-Rich Application: Display Orientation – Anchor Views, resizing and repositioning Views, Managing changes to Screen Orientation; Notifications; Action bar; Dialog box; Adapters- Array Adapters and Base Adapters; Recycler View.

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

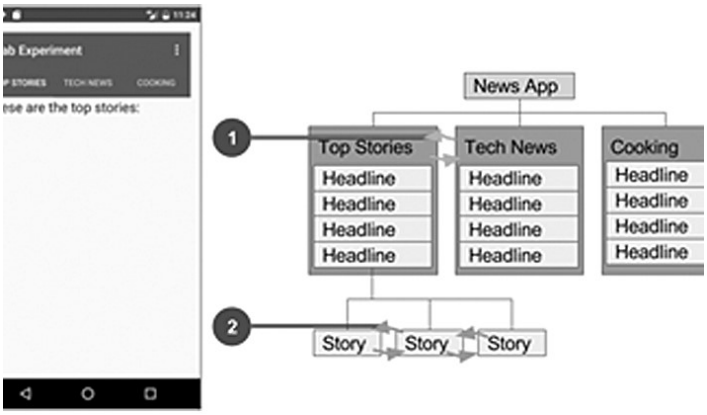
SQLITE DATABASE

SQLite Database – Creating the database, Dealing with CRUD;

Firebase-Getting Started with Firebase, Add Firebase to your Android project, Firebase database-Introduction to Firebase database, set up Firebase Real-time Database for Android, Read and Write Data on Android; Publish the App in Play store.

PRACTICES:

- Design an application to keep data when the user rotates the device, and when the screen is rotated: When the user rotates the device, Android will normally destroy and re-create the current Activity. You want to keep some data across this cycle, but all the fields in your Activity are lost during it.
- Create a Splash Screen for the existing project- Droid Café from Module-1.
- Design a News App- Consider the following screen as reference:



NOTE: Use Recycle View to display the news under each category.

- Adding more features to Droid Café: In the previous assignments, you created an app called Droid Café, using the Basic Activity template. This template also provides a skeletal options menu in the app bar at the top of the screen.
  a) Update that menu option as shown in the following images:
  b) Add notification option: The app must notify the user when the user places the order.
- Provide user authentication for the Droid Café using Firebase – Authentication or SQLite.

- Save all the user preferences in the Firebase Real time Database to fetch whenever required.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply views, intents, and fragments to an existing application.	Apply	1	2
2	Evaluate an existing app to enrich it with new features.	Evaluate	2	2,3
3	Analyse methods for storing, sharing, and retrieving data in an Android app.	Analyse	2	5
4	Design and publish a mobile app in the play store with a database forgiven real-time scenarios using modern tools- Android Studio, and Firebase.	Create	2	5,10

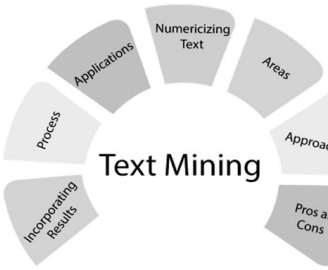
TEXT BOOKS:

1. John Horton “Android Programming for Beginners: Build in-depth, full-featured Android apps starting from zero programming experience”, 3rd Edition, 2021.
2. Wei-Meng Lee, “Beginning Android Application Development”, 1st Edition, John Wiley & Sons, 2012.

REFERENCE BOOKS:

1. Michael Burton,” Android App Development for Dummies “, 3rd Edition, A Wiley Brand, 2020.
2. Dawn Griffiths & David Griffiths, “Headfirst Android Development A Brain-Friendly Guide” 2nd Edition, O’Reilly, 2015.
3. https://aws.amazon.com/mobile/mobile-application-development/
4. https://google-developer-training.github.io/android-developer-fundamentals-course-concepts/.

22CS811 TEXT MINING



Source: https://www.javatpoint.com/text-data-mining

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Data analysis skills, Database, Machine learning or Deep learning algorithms..

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the fundamental concepts and techniques used in Text Processing. Students will gain an in-depth understanding of the computational properties of natural languages and the commonly used algorithms for processing linguistic information. The course examines models and algorithms used in both the traditional symbolic and the more recent statistical approaches.

MODULE-1

UNIT-1 **10L+0P+6P=16 Hours**

INTRODUCTION

NLP tasks in syntax, semantics, and pragmatics. Applications such as information extraction, question answering, and machine translation. The role of machine learning. Brief history of the field.

Regular Expressions, Text Normalization, Edit Distance: Regular Expressions, words, Corpora, Text Normalization and Minimum edit distance.

UNIT-2 **14L+0P+10P=24 Hours**

N-GRAMS, VECTOR SEMANTICS AND EMBEDDING

N-Gram Language Models: N-Grams, Evaluating Language Models, Generalization and Zeros, Smoothing, Kneser-Ney Smoothing, Huge Language Models and Stupid Back off.

Vector Semantics and Embeddings: Lexical semantics, Vector semantics, words and vectors, cosine for measuring similarity, TF-IDF: weighing terms in the vector, Point wise Mutual Information, Applications of TF/ IDP and PPMI vector models, word 2 vec, visualizing embeddings, semantic properties of embeddings, bias and embeddings, evaluating vector models.

PRACTICES:

- Perform basic text pre-processing using the following approaches:
  - Stop word Elimination.
  - Removal of Special Characters.
  - Stemming.
  - Lemmatization.
  - N-gram Tokenization.
- Design and develop a text classification model with various machine learning algorithms over the following feature extraction methods and compare their performance.
  - TF-IDF.
  - Word2 Vec.
- Perform Exploratory Data Analysis using following:
  - Word Cloud.
  - Frequent Word Detection.
  - Keyword Extraction based on ranking.

SKILLS:

- ✓ *Exploratory Data analysis.*
- ✓ *Opinion Mining/ Sentiment Analysis using various deep Learning models.*
- ✓ *Text Summarization and Categorization etc.*

MODULE-2

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

SEQUENCE LABELING FOR PARTS OF SPEECH AND NAMED ENTITIES

Part-of-Speech Tagging, Named Entities and Named Entity Tagging, HMM Part-of-Speech Tagging, Conditional Random Fields, Evaluation of Named Entity Recognition.

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

SEQUENCE AND TRANSLATION PROCESSING

Deep Learning Architectures for Sequence Processing: Language Models Revisited, Self-Attention **Networks:** Transformers.

Machine Translation and Encoder-Decoder Models: The Encoder-Decoder Model, Encoder-Decoder with RNNs, Attention, Beam Search, Encoder-Decoder with Transformers.

PRACTICES:

- Perform Named Entity Recognition to extract required entities from a given unstructured text using NLTK.
- Perform POS Tagging with HMM and also optimize the performance of HMM with Viterbi.
- Design and develop a text classification model using Latent Dirichlet allocation and compare its performance with TF-IDF and Word2Vec.
- Perform text classification using following methods and compare their performance in terms of various evaluation metrics such as Accuracy, Precision, Recall and F-Score.
  - CNN
  - LSTM
  - GRU
  - Encoder-Decoder
  - Transformers
- Design and develop text classification model using Attention.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Understand and apply the usage of regular expressions to process the raw text.	Apply	1	1
2	Apply various approaches to sentiment analysis using Machine Learning methods.	Apply	1,2	1
3	Analyze the vector semantics and embedding in the representation of the text.	Analyze	1	2
4	Design and development various statistical approaches to machine translation.	Evaluate	1,2	3,4,5

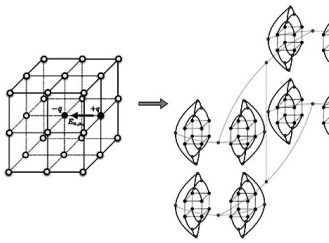
TEXT BOOK:

1. Daniel Jurafsky and James HMartin, “Speech and Language processing: An introduction to Natural Language Processing, Computational Linguistics and speech Recognition”, 3rd Edition, 2020.

REFERENCE BOOKS:

1. Steven Bird, Ewan Klein, Edward Lopper, “Natural Language Processing with Python”, 1st Edition, O’Reilly Publishers, 2009.
2. Nitin Indurkhya, Fred J. Damerau, “Handbook of Natural Language Processing”, 2nd Edition, CRC Publishers, 2010.

22CS812 NUMERICAL ALGORITHMS



Source: https://pasquans.eu/new-numerical-algorithms-for-old-problems/

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Basics of integration, Differentiation and Polynomials.

COURSE DESCRIPTION AND OBJECTIVES:

The goal of this course is to build a base for Numerical methods, which are the basic algorithms underpinning computer predictions in modern systems science. Such methods include techniques for simple optimisation, interpolation from the known to the unknown, linear algebra underlying systems of equations, ordinary differential equations to simulate systems, and stochastic simulation under random influences.

MODULE-1

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

ROOT FINDING METHODS, SYSTEM OF LINEAR EQUATIONS AND INTERPOLATION

Absolute error, order of Convergence, Geometrical Description, method of successive approximation, Bisection method, Regula - Falsi method, Newton's Method. Gauss Seidal method, Crouts method, Triangularization method, Relaxation method.

Interpolation - Finite differences, interpolation, Newton's forward and backward difference formulae, Newton's divided difference and Lagrange's formulae (with proof), errors in interpolation formula.

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

APPLICATIONS

Finding positive, negative and real root of algebraic and transcendental equation, Solution of simultaneous linear algebraic equation. nth difference of a polynomial, finding missing terms in a sequence, sum of n terms in a series, finding polynomial using a given set of data, estimated values of a function inside and outside the given intervals of data.

PRACTICES:

- Finding positive, negative and real root of algebraic and transcendental equation using any of your known programming knowledge.
- Solution of simultaneous linear algebraic equation equation using discrete methods and implement using C.
- By python program estimate the values of a function inside and outside the given intervals of data.

MODULE-2

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

NUMERICAL DIFFERENTIATION, INTEGRATION AND DIFFERENTIAL EQUATION

Numerical Differentiation: Newton's forward and Backward formulas to compute up to second order differentiation of a function.

Numerical Integration: Trapezoidal and Simpson's 1/3 and 3/8 rules.

ODE: Picard's approximation, Milne's Predictor Corrector formulas.

PDE: Liebman's Iteration Process, Bender Schmidth.

SKILLS:

- ✓ *Analyze the types and occurrence of roots.*
- ✓ *Interpolate the unknown values of function.*
- ✓ *Develop a difference equation.*
- ✓ *Gain the knowledge to solve an ODE numerically.*
- ✓ *Gain the knowledge to solve an ODE numerically.*

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

APPLICATIONS

Finding maxima and minima of a function, population growth, acceleration, area bounded by the curve, Solution of ODE, Solution of Elliptic, Parabolic and Hyperbolic PDE.

PRACTICES:

- Finding maxima and minima of a function.
- Solve ODE numerically and plot the curve.
- Classify the PDE.
- Solve PDE numerically and plot the curve.
- Developing difference equations from ODE and PDE.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply numerical methods to find roots.	Apply	1	1, 2, 4, 5, 9, 10, 12
2	Evaluate the unknown values using interpolation.	Apply	1	1, 2, 5, 9, 10
3	Develop a finite difference scheme.	Analyze	2	1, 2, 3, 5, 9, 10
4	Apply numerical methods to solve ODE and PDE and analyse graphically.	Analyze	2	1, 2, 5, 9, 10, 12

TEXT BOOKS:

1. S. S. Sastry, “Introductory methods of numerical analysis”, 5th ed, PHI learning, 2012.
2. M K Jain, “Numerical Methods for Scientific and Engineering Computation”, New Age internarnational, 8th Ed. 2022.

REFERENCE BOOKS:

1. P Kandasamy, “Numerical Methods”, S Chand, 2nd ed, 2015.
2. C. Ray Wylie, Louis C. Barrett: “Advanced Engineering Mathematics”, 6” Edition, 2. McGraw-Hill Book Co., New York, 1995.
3. Gupta C.B., Singh S.R. and Mukesh Kumar: “Engineering Mathematics for Semester I & II”, Mc-Graw Hill Education (India) Pvt. Ltd., 2015.

22CS813 OPERATING SYSTEM DESIGN

Source: https://www.tutorialspoint.com.cach3.com/operating-system-design-goals.html

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Knowledge of computers fundamentals, Computer organization and Digital logic design..

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. Further, it also helps students to understand the different scheduling policies, process synchronization mechanisms, deadlock handling mechanisms and memory management techniques.

MODULE-1

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

INTRODUCTION & PROCESSES

Introduction to Operating System: Introduction, Operating System Concepts, System Calls, Operating System Structure.

Processes: Introduction, Inter Process Communication, Classical IPC Problems, Scheduling, Implementation of MINIX 3, System Task in MINIX 3 and Clock Task in MINIX 3.

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

INPUT/OUTPUT

Input/Output: Principles of Hardware and Software, Deadlock- Principles, Ostrich Algorithm, Detection and Recovery, Deadlock Prevention and Avoidance; Input/output in MINIX 3, Block Devices in MINIX 3.

PRACTICES:

- Implementation of new process creation and its communications.
- Implement of thread creation and deletion.
- Implementation of FCFS scheduling.
- Implementation of SJF and RR Scheduling.
- Implementation of producer consumer problem.
- Implementation of Banker's algorithm for Deadlock avoidance.
- A MINIX file whose owner has UID = 12 and GID = 1 has mode rwxr-x - Another user with UID = 6, GID = 1 tries to execute the file. What will happen?
- In MINIX 3 if user 2 links to a file owned by user 1, then user 1 removes the file, what happens when user 2 tries to read the file?
- Write a program (or series of programs) to test all the MINIX 3 system calls. For each call, try various sets of parameters, including some incorrect ones, to see if they are detected.
- Suppose that a computer can execute 1 billion instructions/sec and that a system call takes 1000 instructions, including the trap and all the context switching. How many system calls can the computer execute per second and still have half the CPU capacity for running application code?
- Write a shell script that produces a file of sequential numbers by reading the last number in the file, adding 1 to it, and then appending to the file. Run one instance of the script in the background and one in the foreground, each accessing the same file. How long does it take before a race condition manifests itself? What is the critical section? Modify the script to prevent the race (Hint: use ln file file.lock to lock the data file).

SKILLS:

- ✓ *Know the concepts of Processes scheduling and File Systems.*
- ✓ *Identification of different disk scheduling methodologies.*
- ✓ *Interpret UNIX Commands, Shell basics, and shell environments*

- Show how counting semaphores (i.e., semaphores that can hold an arbitrarily large value) can be implemented using only binary semaphores and ordinary machine instructions.
- A fast food restaurant has four kinds of employees: (1) order takers, who take customer's orders; (2) cooks, who prepare the food; (3) packaging specialists, who stuff the food into bags; and (4) cashiers, who give the bags to customers and take their money. Each employee can be regarded as a communicating sequential process. What form of interprocess communication do they use? Relate this model to processes in MINIX 3.
- Five batch jobs A through E, arrive at a computer center at almost the same time. They have estimated running times of 10, 6, 2, 4, and 8 minutes. Their (externally determined) priorities are 3, 5, 2, 1, and 4, respectively, with 5 being the highest priority. For each of the following scheduling algorithms, determine the mean process turnaround time. Ignore process switching overhead.

  a) Round robin.

  b) Priority scheduling.

  c) First-come, first-served (run in order 10, 6, 2, 4, 8).

  d) Shortest job first.

  For (a), assume that the system is multi programmed, and that each job gets its fair share of the CPU. For (b) through (d) assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound.
- Solve the dining philosopher's problem using monitors instead of semaphores.
- The banker's algorithm is being run in a system with m resource classes and n processes. In the limit of large m and n, the number of operations that must be performed to check a state for safety is proportional to ma and nb. What are the values of a and b?

MODULE-2

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

MEMORY MANAGEMENT

Memory Management: Basic concept of memory management, Swapping, Virtual Memory, Page Replacement Algorithms, Design issues for Paging systems, Segmentation, Implementation of MINIX 3 Process Manager.

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

FILE SYSTEMS

File Systems: Files, Directories, File System Implementation- layout, Implementing files, Disk Space Management, File System Reliability, File system Performance, Security, Implementation of MINIX 3 File System.

PRACTICES:

- Consider a swapping system in which memory consists of the following hole sizes in memory order: 10 KB, 4 KB, 20 KB, 18 KB, 7 KB, 9 KB, 12 KB, and 15 KB. Which hole is taken for successive segment requests of (a) 12 KB (b) 10 KB (c) 9 KB for first fit? Now repeat the question for best fit, worst fit, and next fit.
- A machine has a 32-bit address space and an 8 KB page. The page table is entirely in hardware, with one 32-bit word per entry. When a process starts, the page table is copied to the hardware from memory, at one word every 100 nsec. If each process runs for 100 msec (including the time to load the page table), what fraction of the CPU time is devoted to loading the page tables?
- A computer with a 32-bit address uses a two-level page table. Virtual addresses are split into a 9-bit top-level page table field, an 11-bit second-level page table field, and an offset. How large are the pages and how many are there in the address space?
- Suppose that a 32-bit virtual address is broken up into four fields, a, b, c, and d. The first three are used for a three-level page table system. The fourth field, d, is the offset. Does the number of pages depend on the sizes of all four fields? If not, which ones matter and which ones do not?

- A computer has four page frames. The time of loading, time of last access, and the R and M bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last ref.	R	M
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	160	280	1	1

  Which page will NRU replace?

  Which page will FIFO replace?

  Which page will LRU replace?

  Which page will second chance replace?

- A small computer has four page frames. At the first clock tick, the R bits are 0111 (page 0 is 0, the rest are 1). At subsequent clock ticks, the values are 1011, 1010, 1101, 0010, 1010, 1100, and 0001. If the aging algorithm is used with an 8-bit counter, give the values of the four counters after the last tick.
- Free disk space can be kept track of using a free list or a bitmap. Disk addresses require D bits. For a disk with B blocks, F of which are free, state the condition under which the free list uses less space than the bitmap. For D having the value 16 bits, express your answer as a percentage of the disk space that must be free.
- A disk has 4000 cylinders, each with 8 tracks of 512 blocks. A seek takes 1 msec per cylinder moved. If no attempt is made to put the blocks of a file close to each other, two blocks that are logically consecutive (i.e., follow one another in the file) will require an average seek, which takes 5 msec. If, however, the operating system makes an attempt to cluster related blocks, the mean interblock distance can be reduced to 2 cylinders and the seek time reduced to 100 microsec. How long does it take to read a 100 block file in both cases, if the rotational latency is 10 msec and the transfer time is 20 microsec per block?
- Write a pair of programs, in C or as shell scripts, to send and receive a message by a covert channel on a MINIX 3 system. Hint: A permission bit can be seen even when a file is otherwise inaccessible, and the sleep command or system call is guaranteed to delay for a fixed time, set by its argument. Measure the data rate on an idle system. Then create an artificially heavy load by starting up numerous different background processes and measure the data rate again.
- Implement immediate files in MINIX 3, that is small files actually stored in the i-node itself, thus saving a disk access to retrieve them.
- Assume that you have a page-reference string for a process with m frames (initially all empty). The page-reference string has length p, and n distinct page numbers occur in it.

  a) What is a lower bound on the number of page faults?

  b) What is an upper bound on the number of page faults?

- Consider the following page-replacement algorithms. Rank these algorithms on a five- point scale from “bad” to “perfect” according to their page-fault rate. Separate those algorithms that suffer from Belady’s anomaly from those that do not.

  a) LRU replacement.

  b) FIFO replacement.

  c) Optimal replacement.

  d) Second-chance replacement.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Classify the basic concepts of operating system.	Analyze	1	1
2	Apply the concepts of process scheduling algorithms and process synchronization techniques to derive the efficiency of resource utilization.	Apply	1	1, 2, 3, 5, 12
3	Synthesize the concepts of I/O management, file system implementation and problems related to Deadlock.	Analyze	1,2	1,2,12
4	Design the various memory management schemes for a given scenario.	Create	2	3,5
5	Apply the concepts of file system interface and implementation.	Apply	2	2,5

TEXT BOOKS:

1. Andrew S. Tanenbaum, Albert S.Woodhull. The MINIX Book Operating Systems Design and Implementation, Pearson Publicastion, 3rd Edition, June 2006.
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, 9th edition, John Wiley & Sons Inc, 2013.

REFERENCE BOOKS:

1. Richard. Stevens and Stephen A Rago, “Advanced Programming in the Unix Environment”, 3rd edition, Addison-Wesley, 2013.
2. William Stallings, “Operating Systems-Internals and Design principles” PHI, 7th Edition, 2012.
3. Gary J. Nutt. Addison-Wesley , “Operating Systems: A Modern Perspective”, Aug 2001, 2nd Edition.
4. B.A. Forouzan& R.F. Giberg, —Unix and shell Programmingll, Thomson, First Edition, New Delhi, 2003.

22CS814 OPTIMIZATION TECHNIQUES

Source: https://oworkers.com/8-process-optimization-techniques-how-to-get-started/

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Probability & Linear algebra.

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the fundamental concepts of Optimization Techniques and to make the learners aware of the importance of optimizations in real scenarios. The most important objective is to provide the concepts of various classical and modern methods for constrained and unconstrained problems in both single and multivariable. Finally, the basic idea behind the evolutionary algorithms like Ant Colony Optimization, Particle Swarm Optimization could be discussed for further study.

MODULE-1

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

INTRODUCTION TO CLASSICAL METHODS AND LINEAR PROGRAMMING PROBLEMS TERMINOLOGY

Design Variables – Constraints, Objective Function - Problem Formulation, Linear Programming Problem - Simplex method - Concept of Duality.

General Transportation Problem: The transportation table finding in initial basic feasible solution, North-West corner method, Least cost method, Row minima method, Column minima method.

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

SINGLE VARIABLE OPTIMIZATION

Problems Optimality Criterion - Bracketing Method - Region Elimination Methods -Interval Halving Method - Fibonacci Search Method - Golden Section Method. Gradient Based Methods: Newton -Raphson Method - Bisection Method - Application to Root finding.

Multi Variable Optimization, Algorithms Optimality: Criteria - Unidirectional Search. Direct Search Methods: Hooke -Jeeves pattern search method. Gradient Based Methods: Cauchy's Steepest Descent Method - Newton's method, Quasi-Newton methods, L-BFGS.

PRACTICES:

A company makes two products (X and Y) using two machines (A and B). Each unit of X that is produced requires 50 minutes processing time on machine A and 30 minutes processing time on machine B. Each unit of Y that is produced requires 24 minutes processing time on machine A and 33 minutes processing time on machine B. At the start of the current week there are 30 units of X and 90 units of Y in stock. Available processing time on machine A is forecast to be 40 hours and on machine B is forecast to be 35 hours. The demand for X in the current week is forecast to be 75 units and for Y is forecast to be 95 units. Company policy is to maximise the combined sum of the units of X and the units of Y in stock at the end of the week. Formulate the problem of deciding how much of each product to make in the current week as a linear program. Solve this linear program graphically.

SKILLS:

- ✓ *Project scheduling process skills including defining project activities, and estimation of time and resources*
- ✓ *Resource optimization skills to adjust the project schedule as per the demand and supply issues of project resources*
- ✓ *Feasibility Analysis.*

Solve using the Simplex method the following problem:

Maximize Z = f(x,y) = 3x + 2y

subject to: $2x + y \leq 18$

$2x + 3y \leq 42$

$3x + y \leq 24$

$x \geq 0 , y \geq 0$

Luminous lamps have three factories - F1, F2, and F3 with production capacity 30, 50, and 20 units per week respectively. These units are to be shipped to four warehouses W1, W2, W3, and W4 with requirement of 20, 40, 30, and 10 units per week respectively. The transportation costs (in Rs.) per unit between factories and warehouses are given below. Find an initial basic feasible solution of the given transportation problem using northwest corner rule.

Factory	**Warehouse**				**Supply**
	W1	W2	W3	W4	
F1	1	2	1	4	30
F2	3	3	2	1	50
F3	4	2	5	9	20
Demand	20	40	30	10	

- A mobile phone manufacturing company has three branches located in three different regions, say Jaipur, Udaipur and Mumbai. The company has to transport mobile phones to three destinations, say Kanpur, Pune and Delhi. The availability from Jaipur, Udaipur and Mumbai is 40, 60 and 70 units respectively. The demand at Kanpur, Pune and Delhi are 70, 40 and 60 respectively. The transportation cost is shown in the matrix below (in Rs). Use the Least Cost method to find a basic feasible solution (BFS).

		DESTINATIONS			
		Kanpur	**Pune**	**Delhi**	**Supply**
Sources	Jaipur	4	5	1	40
	Udaipur	3	4	3	60
	Mumbai	6	2	8	70
	Demand	**70**	**40**	**60**	**170**

- Find Solution using Row minima method.

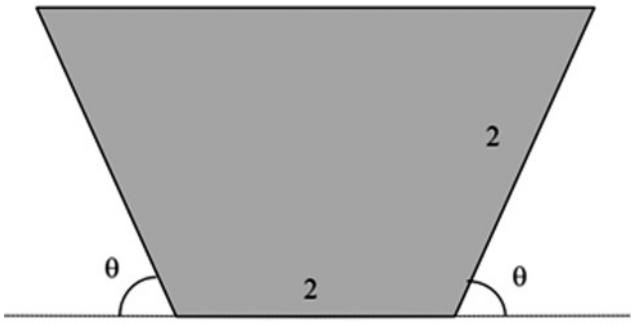
	D1	**D2**	**D3**	**D4**	**Supply**
S1	11	13	17	14	250
S2	16	18	14	10	300
S3	21	24	13	10	400
Demand	200	225	275	250	

- Find Solution using Column minima method.

	D1	D2	D3	D4	Supply
S1	11	13	17	14	250
S2	16	18	14	10	300
S3	21	24	13	10	400
Demand	200	225	275	250	

- Find the positive root of the following equation by (Bisection) method,
  $f(x)=\sin(x/2\pi)-\cos^2 x$
- Consider figure below. The cross-sectional area A of a gutter with equal base and edge length of 2 is given by:
  $A = 4\sin\theta\,(1+\cos\theta)$

Using an initial interval of $[0,\pi/2]$, find the interval after 3 iterations. Use an initial interval $\varepsilon = 0.2$.



MODULE-2

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

STOCHASTIC OPTIMIZATION PROBLEMS

Notion of regret, online to batch conversion, Methods offering vanishing regret - OGD, EG, OMD Convex Sets and Functions: Affine and convex sets, convexity preserving operations, separating and supporting hyper-planes, generalized inequalities, Operations preserving convexity, conjugate function, Quasi-convex functions, Log-concave and log-convex functions, Convexity with respect to generalized inequalities

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

NON-CONVEX OPTIMIZATION PROBLEMS

Applications - sparse recovery, affine rank minimization, low-rank matrix completion, Convex approaches - relaxation-based methods, Non-convex approaches - projected gradient descent, alternating minimization

Algorithms: Unconstrained problems, equality constrained problems

PRACTICES:

- Show that each convex cone is indeed a convex set.
- Consider the linear system $0 \le x_i \le 1$ for $i = 1, \ldots, n$ and let P denote the solution set. Explain how to solve a linear programming problem $\max\{c^T x : x \in P\}$.
  What if the linear system was $a_i \le x_i \le b_i$ for $i = 1, \ldots, n$. Here we assume $a_i \le b_i$ for each i.
- Can projected gradient descent (PGD) be used to obtain a stationary solution?
- Show that strong smoothness does not imply convexity by constructing a nonconvex function $f : R$
- $p \rightarrow R$ that is 1-SS.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Understand the classical, linear programming and transportation problem along with game theory.	Understand	1	1
2	Analyze real world problems around you.	Analyze	1	1, 2
3	Apply optimization to real-world problems.	Apply	1	1, 2, 3
4	Implement dynamic programming in real world complex problems.	Implement	2	3, 4, 11, 12

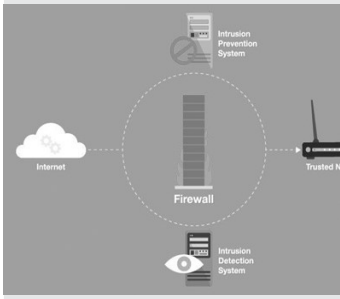
TEXT BOOKS:

1. Kanti Swarup, Man Mohan and P.K.Gupta, “Operations Research”, Sultan Chand &Sons, 2005.
2. S. S. Rao, “Engineering Optimization Theory and Practice”, 4th Edition, Wiley Publishers, 2009.

REFERENCE BOOKS:

1. M.C. Bhuvaneswari, “Application of Evolutionary Algorithms for Multi-Objective Optimization in VLSI and Embedded Systems”, Spinger, 2014.
2. Ashlock D, “Evolutionary Computation for Modeling and Optimization”, Springer, 2006.
3. Kalyanmoy Deb, “Optimization for Engineering Design”, Prentice Hall of India, 2nd Edition, 2012.
4. A. Ravindran and K.M. Rogsdell, G.V. Reklaites, “Engineering Optimization: Methods and Applications”,Wiley, 2nd Edition, 2006.
5. E. J. Haug and J.S. Arora, “Applied Optimal Design”, Wiley Publishers,1979.

22CS815 INTRUSION DETECTION AND PREVENTION SYSTEM



Source: https://www.kirkpatrickprice.com/blog/idps-techniques

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Cryptography and Network Security.

COURSE DESCRIPTION AND OBJECTIVES:

This course aims to understand modern concepts related to Intrusion Detection System. The course compares alternative tools and approaches for Intrusion Detection through quantitative analysis to determine the best tool or approach to reduce risk from intrusion Students undergoing this course can identify and describe the parts of all intrusion detection systems and characterize new and emerging IDS technologies according to the basic capabilities all intrusion detection systems share.

MODULE-1

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

INTRODUCTION

Basic Concepts of Security, Introduction to Intrusions, Need of Intrusion Detection, Classification of Intrusion Detection Systems, Sources of Vulnerabilities, Attacks against various security objectives, countermeasures of attacks.

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

INTRUSION DETECTION AND PREVENTION TECHNOLOGIES

Host-based intrusion detection system (HIDS), Network-based IDS, Information Sources for IDS, Host and Network Vulnerabilities and Countermeasures. Intrusion detection techniques, misuse detection: pattern matching, rule-based and state-based anomaly detection: statistical based, machine learning based, data mining-based hybrid detection.

IDS and IPS Architecture: Tiered architectures, Single-tiered, Multi-tiered, Peer-to-Peer. Sensor: sensor functions, sensor deployment and security.

PRACTICES:

- Installing Snort into the Operating System
- Configuring and Starting the Snort IDS
- Writing and Adding a Snort Rule
- Triggering an Alert for the New Rule

MODULE- 2

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

AGENTS

Agent functions, agent deployment and security. Manager component: manager functions, manager deployment and security. Information flow in IDS and IPS, defending IDS/IPS, Case study on commercial and open-source IDS.

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

ALERT MANAGEMENT AND CORRELATION DATA FUSION

Alert correlation, Pre-process, Correlation Techniques, Post-process, Alert Correlation architectures. Cooperative Intrusion Detection, Cooperative Discovery of Intrusion chain, Abstraction-based Intrusion

SKILLS:

- ✓ *Installation and configuration of IDS.*
- ✓ *Compare various anomaly detection techniques.*
- ✓ *Evaluate security of intrusion detection tool.*

Detection, Interest-based communication and cooperation, agent-based cooperation.

PRACTICES:

- Demo of Eavesdropping attack and its Prevention using SSH.
- Demonstrate how to provide secure data storage, secure data transmission and for creating digital signatures (GnuPG)
- Setup a honey pot and monitor the honeypot on network (KF Sensor)
- Installation of rootkits and study about the variety of option.
- Perform wireless audit on an access point or a router and decrypt WEP and WPA. (NetStumbler)
- Install any open-source IDS and study the logs

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply Intrusion Detection tools and techniques in order to improve their security posture.	Apply	2	1,2
2	Apply the knowledge to the architecture, configuration, and analysis of specific intrusion detection systems.	Apply	2	1, 2
3	Analyse appropriate situations and scenarios where intrusion detection may be applied to achieve an increased level of situational awareness and information assurance.	Analyse	1	1,2
4	Evaluate the security of an organization for better performance.	Evaluate	1	1,2

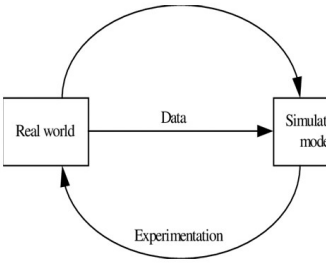
TEXT BOOKS:

1. C. Endorf, E. Schultz and J. Mellander, Intrusion Detection & Prevention, McGraw-Hill/Osborne , 2006.

REFERENCE BOOKS:

1. Ali A. Ghorbani, Network intrusion detection and prevention concepts and techniques, Springer, 2010.
2. J. M. Kizza, Computer Network Security, Springer, 2005.
3. Chris Sanders and Jason Smith, Applied Network Security Monitoring: Collection, Detection, and Analysis, Syngress, 2013.

22CS816 SIMULATION AND MODELLING



Source: https://www.researchgate.net/figure/The-Simulation-Modelling-Process-Simple-Outline_fig1_3834396

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Probability and statistics, OOPS.

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces discrete-event simulation techniques including model design and development, comparison to analytical models, input data preparation, random number generation, output statistical analysis, and model validation and evaluate the performance of real-world systems by analyzing the output of the model under various conditions.

MODULE-1

UNIT-1 **8L+0T+6P=14 Hours**

INTRODUCTION TO SIMULATION, GENERAL PRINCIPLES

When simulation is the appropriate tool and when it is not appropriate, Advantages and disadvantages of Simulation, Areas of application, Systems and system Environment, Components of a System-Discrete and continuous systems, Model of a system, Types of Models, Discrete-Event System Simulation, Steps in a Simulation Study, Simulation Examples.

General Principles, Simulation Software: Concepts in Discrete-Event Simulation, The Event-Scheduling/Time-Advance Algorithm, World Views, Manual simulation Using Event Scheduling, List processing, Basic properties, Operations-Using Arrays, Dynamic Allocation, Linked Lists-Simulation in Java -Simulation in GPSS.

UNIT-2 **16L+0T+10P=26 Hours**

MATHEMATICAL AND STASTICAL MODELS

Statistical Models in Simulation: Review of terminology, concepts, Useful statistical models, Discrete Distributions, Continuous Distributions, Poisson Process, Empirical distributions.

Queuing Models: Characteristics of queuing Systems, Queuing notation, Long-run measures of performance of queuing Systems, Steady-state behavior of M/G/1 queue, Networks of queues, Rough-cut modeling: An illustration

PRACTICES:

- Discuss the system components of a grocery store, healthcare system, material dispatching system.
- Discuss the steps in simulation study for the above given systems.
- A simulation is to be conducted of cooking a porridge to discover at what time a person should start order to have the it on the table by 7:00 P.M. Read a recipe for preparing a dinner or ask a friend or relative for the recipe. As best you can, trace what you understand to be needed, in the data-collection phase of the simulation process, include each step in the recipe. What are the events, activities, and state variables in this system?
- The daily demand for a product is found to follow the distribution as

SKILLS:

- ✓ *Interpret the model and apply the results to resolve critical issues in a real world environment.*
- ✓ *Analyse the Simulation models using input analyzer, and output analyzer.*
- ✓ *Verify and Validate of simulation model.*

Demand	Probability
10	0.25
11	0.35
12	0.30
13	0.10

Determine the total demand for the next 10 days.

- Students are arriving at the college office at the rate of one every 6 ± 2 minutes to pay the fees. They hand over the forms to one of the two clerks available and it takes l 0 ± 2 minutes for the clerk to verify each form. Then the forms are sent to a single cashier who takes 6 ± l minute per form. Simulate the system for 100 hours and determine the

  (a) utilization of each clerk (b) utilization of the cashier (c) average time required to process a form (clerk + cashier).

- A patient arrives at the Emergency Room at Hello-Hospital about every 40 ± 19 minutes. Each patient will be treated by either Doctor Slipup or Doctor Gutcut. Twenty percent of the patients are classified as NIA (need immediate attention) and the test as CW (can wait). NIA patients are given the highest priority (3), see a doctor as soon as possible for 40 ± 37 minutes, but then iheir priority is reduced to 2 and they wait until a doctor is free again, when they receive further treatment for 30 ± 25 minutes and are then discharged. CW patients initially receive the priority 1 and are treated (when their torn comes) for 15 ± 14 minutes; their priority is then increased to 2, they wait again until a doctor is free and receive l 0 ± 8 minutes of final treatment, and are then discharged. Simulate for 20 days of continuous operation, 24 hours per day. Precede this by a 2-day initialization period to load the system with patients. Report conditions at times 0 days, 2 days, and 22 days. Does a 2-day initialization appear long enough to load the system to a level reasonably close to steady-state conditions? (a) Measure the average and maximum queue length of NIA patients from anival to first seeing a doctor. What percent do not have to wait at all? Also tabulate and plot the distribution of this initial waiting time for NIA patients. What percent wait less than 5 minutes before seeing a doctor? (b) Tabulate and plot the distribution of total time in system for all patients. Estimate the 90% quantile-that is, 90% of the patients spend less than x amount of time in the system. Estimate x. (c) Tabulate and plot the distribution of remaining time in system from after the first treatment to discharge, for all patients. Estimate the 90% quantile. (Note: Most simulation packages provide the facility to automatically tabulate the distribution of any specified variable).

- Using Excel, generate 12 columns, each with 250 values, using the formula = RAND(). In cell Ml, place the formula = SUM(Al:Ll)-6 and copy it to the 249 cells below M1 in column M.

a) Compute descriptive statistics about the data in that column, including minimum value, maximum value, mean, median, and standard deviation.

b) Tabulate the values with 9 bins: the first bin will include all values less than or equal to -3.5; the next six bins are of width one; the last bin will include all values greater than 3.5.

c) Does the histogram resemble any distribution with which you are familiar? If so, what is its name? Hint l: Use FREQUENCY in Excel to form bins.

- Of the orders a job shop receives, 25% are welding jobs and 75% are machining jobs. What is the probability that (a) half of the next five jobs will be machining jobs? (b) the next four jobs will be welding jobs?
- Students' arrival at a university library follows Poisson with a mean of 20 per hour. Determine (a) the probability that there are 50 arrivals in the next l hour. (b) the probability that no student arrives in the next l hour. (c) the probability that there are 75 arrivals in the next 2 hours.
- The cars arriving at a gas station is Poisson distributed with a mean of lO per minute. Determine the number of pumps to be installed if the ftrm wants to have 50% of arriving cars as zero entries (i.e., cars serviced without waiting).
- Given the following distributions, Normal (10, 4) Triangular (4, 10, 16) Uniform (4, 16) find the probability that 6 < X< 8 for each of the distributions.
- Vehicles pass through a toll gate at a rate of 90 per hour. The average time to pass through the gate is 36 seconds. The arrival rate and service rate follow Poisson distribution. There is

a complaint that the vehicles wait for a long duration. The authorities are willing to install one more gate to reduce the average time to pass through to 30 seconds, if the idle time of the toll gate is less than 10% and the present average queue length at the gate is more than five vehicles. Check whether the installation of the second gate is justified.

SKILLS:

- ✓ *Interpret the model and apply the results to resolve critical issues in a real world environment.*
- ✓ *Analyse the Simulation models using input analyzer, and output analyzer.*
- ✓ *Verify and Validate of simulation model.*

MODULE-2

UNIT-1 **6L+0T+10P=26 Hours**

RANDOM NUMBERS AND INPUT MODELLING

Random-Number Generation, Random-Variate Generation: Properties of random numbers, Generation of pseudo-random numbers, Techniques for generating random numbers, Tests for Random Numbers, Random- Variate Generation, Inverse transform technique, Acceptance-Rejection technique, Special properties.

Input Modeling: Data Collection, Identifying the distribution with data, Parameter Estimation, Goodness of Fit Tests, Fitting a non-stationary Poisson process, Selecting input models without data, Multi-variate and Time-Series input models.

UNIT-2 **8L+0T+6P=14 Hours**

ANALYSIS OF SIMULATION DATA

Verification and Validation of simulation models: Optimization of simulation Models: Model Building, Verification, Validation, Verification of simulation models, Calibration, Validation of models, Optimization, Optimization via Simulation

Output Analysis for A Single Model: Types of simulations with Respect to Output analysis, Stochastic Nature of Output Data, Measures of Performance and their Estimation, Output Analysis for Terminating Simulations, Output analysis for steady-State Simulations.

PRACTICES:

- Develop the triangular random-variate generator with range (0, 12) and mode 5.
- In an inventory system, the lead time is found to follow uniform distribution with mean 10 days and half width 3 days. Generate five lead times.
- Write a computer program to generate exponential random variates for a given mean value. Generate 1000 values and verify the variates generated using chi-square test.
- In a college library, collect the following in formation at the books return counter: arrival of students for returning books service time taken by the counter clerk 305 Consolidate the data collected and verify whether it follows any standard distribution.
- Draw the pdf of normal distribution with J.l. = 6, a= 3
- The following data represent the time to perform transactions in a bank, measured in minutes: 0.740, 1.28, 1 .46, 2.36, 0.354, 0.750, 0.912, 4.44, O.l l4, 3.08, 3.24, 1. 10, 1.59, 1.47, 1.17, 1.27, 9. 12, 1 1.5, 2.42, 1.77. Develop an input model for these data.
- A simulation model. of a job shop was developed to investigate diff erent schedulliJg rules. To validate the model, the scheduling rule currently used was incorporated into the model and the resulting output was compared agamst observed system behavior. By searching the previous year's database records it was estimated that the average number of jobs in the shop was 22.5 on a given day. Seven independent replications of the model were run, each of 30 days' duration, with the following results for average number of Jobs in the shop: 18.9 22.0 19.4 22.1 19.8 21.9 20.2 (a) Develop and conduct a stastical test to evaluate whether model output is consistent with system behavior. Use the level of significance a= 0.05. · (b) What is the power of this test if a difference of two jobs is viewed as critical? What sample size is needed to guarantee a power of 0.8 or higher? (Use a= 0.05.)

- Consider the following inventory system: (a) Whenever the inventory level falls to or below 10 units, an order is placed. Only one order can be outstanding at a time. (b) The size of each order is Q; Maintaining an inventory costs $0.50 per day per item in inventory. Placing an order incurs a fixed cost, $10.00. (c) Lead time is distributed in accordance with a discrete uniform distribution between zero and 5 days. (d) If a demand occurs during a period when the inventory level is zero, the sale is lost at a cost of $2.00 per unit. (e) The number of customers each day is given by the following distribution:

No of customers per day	Probability
1	0.23
2	0.41
3	0.22
4	0.14

f) The demand on the part of each customer is Poisson distributed with a mean of 3 units.

g) For simplicity, assume that an·demands occur at noon and that all orders are placed immediately thereafter. Assume further that orders are received at 5:00 P.M., or after the demand that occurred on that day. Consider the poi icy having Q = 20 .. Make five independent replications, each of length 100 days, and compute a 90% confidence interval for long-run mean daily cost. Investigate the effect of initial inventory level and existence of an outstanding order on the estimate of mean daily cost. Begin with an initial inventory of Q + I 0 and no outstanding orders.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze the steps in design of various simulation models.	Analyze	1	2
2	Conceptualize real world situations related to systems development decisions, originating from source requirements and goals. Construct model for a given set of data and motivate its validity.	Apply	1	1, 4
3	Generate and test random number variates and apply them to develop simulation models.	Apply	2	1
4	Analyze output data produced by a model and test validity of the model.	Analyze	2	2

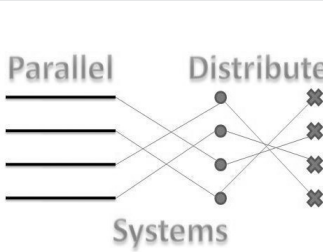
TEXT BOOKS:

1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol, “Discrete-Event System Simulation”, 5th Edition, Pearson Education © 2013.
2. Zeigler B.P. Praehofer. H. and Kim I.G. “Theory of modeling and simulation”, 2nd Edition Academic press 2000.

REFERENCE BOOKS:

1. Averill M. Law, “Simulation Modeling and Analysis”, 4th Edition, Tata McGraw-Hill, 2007.
2. Lawrence M. Leemis, Stephen K. Park, “Discrete – Event Simulation: A First Course”, Pearson Education, 2006.

22CS817 PARALLEL AND DISTRIBUTED COMPUTING



Source: http://uceou.edu/PDS/About%20PDS.html

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Fundamental of programming and operating system, computer networking, and computer Organization..

COURSE DESCRIPTION AND OBJECTIVES:

This course covers the challenges faced in constructing parallel and distributed applications, Various implementation techniques, paradigms, architectures and parallel algorithms. And current trends in parallel and distributed computing like Open MP, POSIX Threads, Apache Hadoop (DFS).

MODULE-1

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

INTRODUCTION TO PARALLEL COMPUTING

The idea of Parallelism, Power and potential of parallelism, examining sequential and parallel programs, Scope and issues of parallel and distributed computing, Goals of parallelism, Parallelism and concurrency using multiple instructions streams.

Parallel Architecture: Pipeline architecture, Array processor, Multi-processor architecture, Systolic architecture, Dataflow architecture, Architectural classification schemes, Memory access classification, Memory Issues: Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, Vector processing, GPU co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).

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

PARALLEL ALGORITHM DESIGN PRINCIPLES AND PROGRAMMING

Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task- Based Decomposition, Data Parallel Decomposition, Characteristics of task and interaction, Load balancing, Data Management, parallel algorithm models, Sources of overhead in parallel programs, Performance metrics for parallel algorithm implementations, Parallel algorithmic patterns like divide and conquer, Map and Reduce, Specific algorithms like parallel Merge Sort, Parallel graph Algorithms.

PRACTICES:

- Identify Multiple Instruction Single Data, or MISD. How would an MISD system work? Give an example.
- Suppose a shared-memory system uses snooping cache coherence and write-back caches. Also suppose that core 0 has the variable x in its cache, and it executes the assignment x = 5. Finally suppose that core 1 doesn't have x in its cache, and after core 0's update to x, core 1 tries to execute y = x. What value will be assigned to y? Why?
- Consider a simplified version of bucket-sort. You are given an array A of n random integers in the range [1...r] as input. The output data consist of r buckets, such that at the end of the algorithm, Bucket i contains indices of all the elements in A that are equal to i.
    - Describe a decomposition based on partitioning the input data (i.e., the array A) and an appropriate mapping onto p processes. Describe briefly how the resulting parallel algorithm would work.
    - Describe a decomposition based on partitioning the output data (i.e., the set of r buckets) and an appropriate mapping onto p processes. Describe briefly how the resulting parallel algorithm would work.

SKILLS:

- ✓ *Recognize parallelism in computational problems.*
- ✓ *Know different parallel systems and their classification.*
- ✓ *Design parallel algorithms for different applications.*
- ✓ *Compare replication schemes with respect to performance, availability and consistency concerns.*
- ✓ *Design, implement, and debug distributed systems.*
- ✓ *Implement parallel algorithms using MPI and OpenMP environments.*
- ✓ *Element parallel algorithms using MPI and OpenMP environments 3.0 3.5*
- ✓ *AVERAGE.*

- Consider seven tasks with running times of 1, 2, 3, 4, 5, 5, and 10 units, respectively. Assuming that it does not take any time to assign work to a process, compute the best- and worst-case speedup for a centralized scheme for dynamic mapping with two processes.

MODULE-2

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

INTRODUCTION TO DISTRIBUTED SYSTEMS

Goals of the Distributed Systems, Relation to parallel systems, synchronous versus asynchronous execution, design issues and challenges, Types of Distributed Systems, Distributed System Models, Hardware and software concepts related to distributed systems, middleware models.

Distributed Computing and Communication design principles: A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication. Communication and Coordination: Shared Memory, Consistency, Atomicity, Message- 08 Passing, Consensus, Conditional Actions, Critical Paths, Scalability, and cache coherence in multiprocessor systems, synchronization mechanism.

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

PARALLEL AND DISTRIBUTED PROGRAMMING FRAMEWORKS

Overview of CUDA, Open MP, POSIX Threads, Apache Hadoop (DFS), and current trends in parallel and distributed computing.

PRACTICES:

- Give five types of hardware resource and five types of data or software resource that can usefully be shared. Give examples of their sharing as it occurs in practice in distributed systems.
- The host computers used in peer-to-peer systems are often simply desktop computers in users' offices or homes. What are the implications of this for the availability and security of any shared data objects that they hold and to what extent can any weaknesses be overcome through the use of replication?
- Consider two communication services for use in asynchronous distributed systems. In service A, messages may be lost, duplicated or delayed and checksums apply only to headers. In service B, messages may be lost, delayed or delivered too fast for the recipient to handle them, but those that are delivered arrive with the correct contents. Describe the classes of failure exhibited by each service. Classify their failures according to their effects on the properties of validity and integrity. Can service B be described as a reliable communication service?
- Illustrate distributed design through a substantial case study, examining in detail the design of the Google infrastructure, a platform and associated middleware that supports both Google search and a set of associated web services and applications including Google Apps.
- Implementation of the parallel algorithms (on a PC-cluster under Linux platform). The programs will be based on POSIX Thread, MPI programming, Hadoop, Apache Spark etc.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply the fundamentals of parallel and parallel computing including architectures and paradigms.	Apply	1	1
2	Analyse the various design principles of parallel algorithms.	Analyse	1	2
3	Learn the intricacies of distributed programming.	Under-stand	2	1
4	Develop and execute basic parallel and distributed applications using basic programming models and tools.	create	2	5

TEXT BOOKS:

1. DISTRIBUTED SYSTEMS Concepts and Design Fifth Edition. George Coulouris. Cambridge University. Jean Dollimore formerly of Queen Mary, University of London.
2. Distributed Systems Principles and Paradigms Andrew S. Tanenbaum Maarten Van Steen, 3rd Edition, 2017.

REFERENCE LINKS:

1. Introduction To Parallel Programming, Peter S. Pacheco University of San Francisco.
2. Introduction To Parallel Processing, M.Sasikumar, Dinesh Shikhare and P. Ravi Prakash, Randy Chow, T. Johnson, Distributed Operating Systems and Algorithms, Addison Wesley.
3. Ian Foster: Designing and Building Parallel Programs – Concepts and tools for Parallel Software Engineering, Pearson Publisher, 1st Edition, 2019.
4. Parallel Programming in C with MPI and Open MP Michael J.Quinn, McGrawHill Higher Education.
5. https://hpc.llnl.gov/training/tutorials/introduction-parallel-computing-tutorial.
6. https://www.geeksforgeeks.org/introduction-to-parallel-computing/.
7. https://nptel.ac.in/.
8. https://www.coursera.org/.

HONOURS

COMPUTER SCIENCE AND ENGINEERING

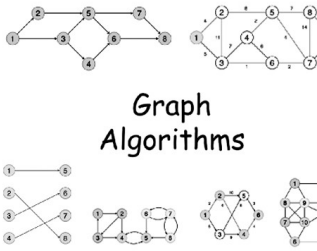
B.Tech.

- 22CS951 - Advanced Graph Algorithms
- 22CS952 - Biometrics
- 22CS953 - Parallel and Distributed Computing
- 22CS954 - Internet of Things
- 22CS955 - Wireless Sensor Networks
- 22CS956 - Capstone Project

COURSE CONTENTS

I SEM & II SEM

22CS951 ADVANCED GRAPH ALGORITHMS



Source: https://towardsdatascience.com/10-graph-algorithms-visually-explained-e57faa1336f3

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Basic Logical Thinking and Problem Solving Ability.

COURSE DESCRIPTION AND OBJECTIVES:

The course will cover some traditional discrete approaches to various graph problems, especially flow problems, and then contrast these approaches with modern, asymptotically faster methods based on combining convex optimization with spectral and combinatorial graph theory.

MODULE-1

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

INTRODUCTION

Introduction to Optimization, Convex Geometry, Linear Algebra, Convexity and Second Derivatives, Gradient Descent and Acceleration.

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

SPECTRAL GRAPH THEORY

Introduction to Spectral Graph Theory, Effective Resistance, Gaussian Elimination as Optimization, Additive Perspective on Gaussian Elimination.

PRACTICES:

- Implement the gradient descent optimization algorithm with Nesterov Momentum.
- Assume that S is subset of R^n is a convex set and that the function f : S->R is convex. Suppose that x1, …, xn belongs to S and theta_1,…,theta_n >= 0 with theta_1,…,theta_n = 1. Prove that f(theta_1x1 + _ _ _ + theta_nxn) <= theta_1f(x1) + …. + theta_nf(xn)

MODULE-2

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

RANDOM MATRIX CONCENTRATION

Introduction to Random Matrix Concentration, Spectral Graph Sparsification, Laplacian Linear Equations, Classical Algorithms for Maximum Flow

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

SEPARATING HYPERPLANE THEOREM

Separating Hyperplane Theorem, Langrange Multipliers, and Convex Duality, Karush-Kuhn-Tucker Conditions, Fenchel Conjugates, Newton's Method.

SKILLS:

- ✓ *Develop a deeper understanding of fundamental phenomena in optimization.*
- ✓ *Deep dive into modern approaches to graph algorithms using convex optimization techniques.*
- ✓ *Central techniques in the development of graph algorithms including graph decomposition techniques, oblivious routing etc.*

PRACTICES:

- Show that the Ford-Fulkerson algorithm may not terminate; moreover, it may converge a value not equal to the value of the maximum flow.
- An electric company is setting up a power plant in a foreign country and it has to plan its capacity. The peak period demand for power is given by p1 = 400 − q1 and the off-peak is given by p2 = 380 − q2. The variable cost to is 20 per unit (paid in both markets) and capacity costs 10 per unit which is only paid once and is used in both periods.
  - Write down the lagrangian and Kuhn-Tucker conditions for this problem
  - Find the optimal outputs and capacity for this problem.
  - How much of the capacity is paid for by each market (i.e. what are the values of λ1 and λ2)?
  - Now suppose capacity cost is 30 per unit (paid only once). Find quantities, capacity and how much of the capacity is paid for by each market (i.e. λ1 and λ2)?

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze key concepts in optimization such as first and second-order optimization, convex duality, multiplicative weights and dual-based methods, acceleration, preconditioning, and non-Euclidean optimization.	Analyze	1	1, 2, 9, 10, 12
2	Design convex optimization through the lens of graph algorithms	Create	1	1,2,3,9
3	Apply the central techniques in the development of graph algorithms including graph decomposition techniques, sparsification, oblivious routing, and spectral and combinatorial preconditioning.	Apply	2	1, 2, 9, 10, 12

TEXT BOOKS:

1. Boyd, Stephen, Stephen P. Boyd, and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.
2. Cartan, Henri. Differential calculus. Hermann, 1983.

REFERENCES:

1. Tarjan, Robert Endre. Data structures and network algorithms. Society for industrial and Applied Mathematics, 1983.
2. Cook, William J., et al. “Combinatorial optimization.” Oberwolfach Reports 5.4 (2009): 2875-2942.
3. Rockafellar, R. Tyrrell. Convex analysis. Vol. 18. Princeton university press, 1970.

22CS952 BIOMETRICS

Source: https://cyberhoot.com/cybrary/biometrics/

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Engineering Mathematics, Probability and Statistics, Image Processing.

COURSE DESCRIPTION AND OBJECTIVES:

This course is a foundation for design and implementation of biometric systems which helps in providing authentication, identification of persons. This can be performed using the biometric traits like finger print, face, iris etc. It is mainly used in security applications.

MODULE-1

UNIT-1 **8L+0T+6P=14 Hours**

INTRODUCTION

Introduction: Biometric Systems, Biometric Functionalities, Biometric System Errors. The Design Cycle of Biometric Systems, Applications of Biometric Systems.

UNIT-2 **16L+0T+10P=26 Hours**

FINGERPRINT AND FACE RECOGNITION SYSTEMS

Finger Print Recognition: Introduction, Friction Ridge Pattern, Fingerprint Acquisition, Feature Extraction, Matching, Fingerprint Indexing, Fingerprint Synthesis, Palmprint.

Face Recognition: Introduction, Image Acquisition, Face Detection, Feature Extraction and Matching.

PRACTICES:

- The minutiae-based representation and matching algorithms.
- Collect various data sets of different traits and analyse the usage of these data sets.
- For a given fingerprint, identify the minutia points by binarization and thinning of ridges.
- From a fingerprint image, extract the singularity points.
- Identify the inter subject and intra subject variations for given finger prints.
- Implementation of ViolaJones object detection algorithm.
- Perform feature extraction by using
  a. Principal component analysis.
  b. Linear Discriminant analysis

MODULE-2

UNIT-1 **10L+0T+8P=18 Hours**

IRIS RECOGNITION

Iris recognition: Introduction, Design of Iris Recognition System, Iris Segmentation, Iris Normalization, Iris Encoding and Matching, Iris Quality.

UNIT-2 **14L+0T+8P=22 Hours**

MULTI-BIOMETRICS

Multi-biometrics: Introduction, Sources of Multiple Evidence, Acquisition and Processing Architecture, Fusion Levels.

SKILLS:

- ✓ *Identify the datasets to be used in various applications.*
- ✓ *Usage of multi-biometrics to enhance security.*
- ✓ *Design of fingerprint, Iris, face detection systems.*

PRACTICES:

- Implementation of Iris segmentation.
- Generation of Iris code.
- Taking a biometric trait, performing.
  a. Sensor-level fusion.
  b. Feature-level fusion.
- Working with feature normalization for the fusion of two heterogeneous feature vectors.
- Score level fusion using various classifiers.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze the steps in design of various biometric system, functionalities, traits and metrics used to measure their performance.	Analyze	1	2
2	Design of biometric systems depending on the choice of using the trait like fingerprint, face, iris etc.	Create	1,2	1, 3, 6
3	Applying the multimodal biometric traits and fusion levels in various applications.	Apply	2	1,6
4	Analyze the usage of various biometric traits in real time applications.	Analyze	2	2,6

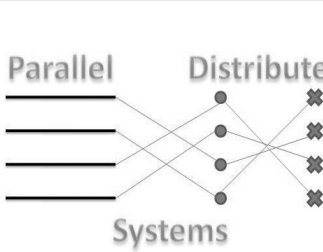
TEXT BOOKS:

1. Anil K. Jain, Arun A. Ross, Karthik Nanda kumar, “Introduction to Biometrics”, Springer, 2011.
2. N. V. Boulgouris , Konstantinos N. Plataniotis , Evangelia Micheli-Tzanakou “Biometrics: Theory, Methods, and Applications”, Wiley, 2009

REFERENCE BOOKS:

1. Samir Nanavati, Michael Thieme, Raj Nanavati, “Biometrics – Identity Verification in a Networked World”, WILEY, 2002.
2. John D. Woodward, John D.Woodward, Jr.Noicholas M.Orlans Peter T. Hig, “Biometrics-The Ultimate Reference”, DreamTech Press, 2003.
3. Julian Ashbourn, “Biometrics: Advanced Identity Verification The Complete Guide” Springer, 2020.

22CS953 PARALLEL AND DISTRIBUTED COMPUTING



Source: http://uceou.edu/PDS/About%20PDS.html

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Fundamental of programming and operating system, computer networking, and computer Organization..

COURSE DESCRIPTION AND OBJECTIVES:

This course covers the challenges faced in constructing parallel and distributed applications, Various implementation techniques, paradigms, architectures and parallel algorithms. And current trends in parallel and distributed computing like Open MP, POSIX Threads, Apache Hadoop (DFS).

MODULE-1

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

INTRODUCTION TO PARALLEL COMPUTING

The idea of Parallelism, Power and potential of parallelism, examining sequential and parallel programs, Scope and issues of parallel and distributed computing, Goals of parallelism, Parallelism and concurrency using multiple instructions streams.

Parallel Architecture: Pipeline architecture, Array processor, Multi-processor architecture, Systolic architecture, Dataflow architecture, Architectural classification schemes, Memory access classification, Memory Issues: Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, Vector processing, GPU co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).

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

PARALLEL ALGORITHM DESIGN PRINCIPLES AND PROGRAMMING

Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task- Based Decomposition, Data Parallel Decomposition, Characteristics of task and interaction, Load balancing, Data Management, parallel algorithm models, Sources of overhead in parallel programs, Performance metrics for parallel algorithm implementations, Parallel algorithmic patterns like divide and conquer, Map and Reduce, Specific algorithms like parallel Merge Sort, Parallel graph Algorithms.

PRACTICES:

- Identify Multiple Instruction Single Data, or MISD. How would an MISD system work? Give an example.
- Suppose a shared-memory system uses snooping cache coherence and write-back caches. Also suppose that core 0 has the variable x in its cache, and it executes the assignment x = 5. Finally suppose that core 1 doesn't have x in its cache, and after core 0's update to x, core 1 tries to execute y = x. What value will be assigned to y? Why?
- Consider a simplified version of bucket-sort. You are given an array A of n random integers in the range [1...r] as input. The output data consist of r buckets, such that at the end of the algorithm, Bucket i contains indices of all the elements in A that are equal to i.
    - Describe a decomposition based on partitioning the input data (i.e., the array A) and an appropriate mapping onto p processes. Describe briefly how the resulting parallel algorithm would work.
    - Describe a decomposition based on partitioning the output data (i.e., the set of r buckets) and an appropriate mapping onto p processes. Describe briefly how the resulting parallel algorithm would work.

SKILLS:

- ✓ *Recognize parallelism in computational problems.*
- ✓ *Know different parallel systems and their classification.*
- ✓ *Design parallel algorithms for different applications.*
- ✓ *Compare replication schemes with respect to performance, availability and consistency concerns.*
- ✓ *Design, implement, and debug distributed systems.*
- ✓ *Implement parallel algorithms using MPI and OpenMP environments.*
- ✓ *Element parallel algorithms using MPI and OpenMP environments 3.0 3.5*
- ✓ *AVERAGE.*

- Consider seven tasks with running times of 1, 2, 3, 4, 5, 5, and 10 units, respectively. Assuming that it does not take any time to assign work to a process, compute the best- and worst-case speedup for a centralized scheme for dynamic mapping with two processes.

MODULE-2

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

INTRODUCTION TO DISTRIBUTED SYSTEMS

Goals of the Distributed Systems, Relation to parallel systems, synchronous versus asynchronous execution, design issues and challenges, Types of Distributed Systems, Distributed System Models, Hardware and software concepts related to distributed systems, middleware models.

Distributed Computing and Communication design principles: A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication. Communication and Coordination: Shared Memory, Consistency, Atomicity, Message- 08 Passing, Consensus, Conditional Actions, Critical Paths, Scalability, and cache coherence in multiprocessor systems, synchronization mechanism.

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

PARALLEL AND DISTRIBUTED PROGRAMMING FRAMEWORKS

Overview of CUDA, Open MP, POSIX Threads, Apache Hadoop (DFS), and current trends in parallel and distributed computing.

PRACTICES:

- Give five types of hardware resource and five types of data or software resource that can usefully be shared. Give examples of their sharing as it occurs in practice in distributed systems.
- The host computers used in peer-to-peer systems are often simply desktop computers in users' offices or homes. What are the implications of this for the availability and security of any shared data objects that they hold and to what extent can any weaknesses be overcome through the use of replication?
- Consider two communication services for use in asynchronous distributed systems. In service A, messages may be lost, duplicated or delayed and checksums apply only to headers. In service B, messages may be lost, delayed or delivered too fast for the recipient to handle them, but those that are delivered arrive with the correct contents. Describe the classes of failure exhibited by each service. Classify their failures according to their effects on the properties of validity and integrity. Can service B be described as a reliable communication service?
- Illustrate distributed design through a substantial case study, examining in detail the design of the Google infrastructure, a platform and associated middleware that supports both Google search and a set of associated web services and applications including Google Apps.
- Implementation of the parallel algorithms (on a PC-cluster under Linux platform). The programs will be based on POSIX Thread, MPI programming, Hadoop, Apache Spark etc.

COURSE OUTCOMES:

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

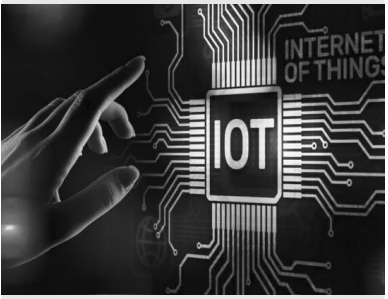
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply the fundamentals of parallel and parallel computing including architectures and paradigms.	Apply	1	1
2	Analyse the various design principles of parallel algorithms.	Analyse	1	2
3	Learn the intricacies of distributed programming.	Under-stand	2	1
4	Develop and execute basic parallel and distributed applications using basic programming models and tools.	create	2	5

TEXT BOOKS:

1. DISTRIBUTED SYSTEMS Concepts and Design Fifth Edition. George Coulouris. Cambridge University. Jean Dollimore formerly of Queen Mary, University of London.
2. Distributed Systems Principles and Paradigms Andrew S. Tanenbaum Maarten Van Steen, 3rd Edition, 2017.

REFERENCE LINKS:

1. Introduction To Parallel Programming, Peter S. Pacheco University of San Francisco.
2. Introduction To Parallel Processing, M.Sasikumar, Dinesh Shikhare and P. Ravi Prakash, Randy Chow, T. Johnson, Distributed Operating Systems and Algorithms, Addison Wesley.
3. Ian Foster: Designing and Building Parallel Programs – Concepts and tools for Parallel Software Engineering, Pearson Publisher, 1st Edition, 2019.
4. Parallel Programming in C with MPI and Open MP Michael J.Quinn, McGrawHill Higher Education.
5. https://hpc.llnl.gov/training/tutorials/introduction-parallel-computing-tutorial.
6. https://www.geeksforgeeks.org/introduction-to-parallel-computing/.
7. https://nptel.ac.in/.
8. https://www.coursera.org/.



Source: https://toolsense.io/glossary/iot/

22CS954 INTERNET OF THINGS

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Computer networks.

COURSE DESCRIPTION AND OBJECTIVES:

Internet of Things (IoT) is presently a hot technology worldwide. Government, academia, and industry are involved in different aspects of research, implementation, and business with IoT. IoT cuts across different application domain verticals ranging from civilian to defense sectors. These domains include agriculture, space, healthcare, manufacturing, construction, water, and mining, which are presently transitioning their legacy infrastructure to support a IoT. To introduce the terminology, technology and its applications, to introduce the concept of M2M (machine to machine) with necessary protocols.

MODULE-1

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

INTERNET OF THINGS FUNDAMENTALS

Introduction to Internet of Things; Physical design & Functional Block of IoT, Device architectures, CoreIoT Functional Stack; Resource constrained devices; Sensors and Components; IoT Enabling Technologies. Societal Benefits of IoT (Domain Specific), Risks, Privacy, and Security.

Network And Communication Protocols: Network Components; Internet Structure, Wireless Protocols; IoT Communication Model & APIs, Wireless Protocol Stack, IoT levels.

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

IOT AND M2M

Software defined networks, Network function virtualization, difference between SDN and NFV for IoT Basics of IoT System Management with NETCONF-YANG: SDN; NFV; Need for IOT Systems Management; SNMP-NETCONF, YANG; IOT Systems management with NETCONF-YANG.

PRACTICES:

- Identify different Sensors and IoT devices
- Identify the Components in Raspberry pi, Arduino, and UNO boards
- Examine IoT levels with any one domain specific application like home automation, weather monitoring system etc.
- Design the Network Configuration and System Management with IoT devices using NETCONF-YANG.
- Design the Network Configuration and System Management with IoT devices using SNMP-NETCONF.

MODULE-2

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

INTRODUCTION TO SYSTEMS DESIGN & DEVELOPMENT

IoT system building blocks, Arduino, Node MCU– Board details, IDE programming; Raspberry Pi-Model and Interfaces, Platform: Axonize, Blynk IoT platform, Fogwing.

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

SKILLS:

- ✓ *Sensor Identification and IoT system design.*
- ✓ *Sensor data analysis.*
- ✓ *Tool usage for developing IoT applications.*

PROGRAMMING AND CASE STUDY

Embedded C vs Python; Operating systems for constrained devices; Domain Specific IoT Application, Task Support IoT Example: The Refrigerator, Weather Monitoring System – Case study- Design, Programming and Execution.

PRACTICES:

- 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.
- Design and Implementation of controlling LED-using Python in Raspberry Pi board.
- Design and Implementation of sensing light through LDR using Python in Raspberry Pi board.
- Design and Implementation to find obstacles through sensor using Python in Raspberry Pi board.
- Design and Implementation of sensing and display temperature using Python in Raspberry Pi board.
- Design and Implementation of detecting noise through microphone sensor using Python in Raspberry Pi board.
- Design and Implementation of output devices through relay using Python in Raspberry Pi board.
- Design and Implementation of vibration sensor using Python in Raspberry Pi board.
- Design and Implementation of uploading sensor data into cloud using Python.

COURSE OUTCOMES:

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

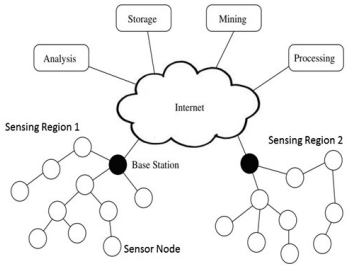
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Illustrate the impact and challenges posed by IoT networks leading to new architectural models.	Analyze	1	4, 6
2	Design an end-to-end Machine-learning model to realize solutions for real-world problems.	Design	1	3
3	Apply various machine-learning models to develop IoT applications.	Apply	2	1
4	Compare and contrast the deployment of smart objects and the technologies to connect them to network.	Evaluate	2	4

TEXT BOOKS:

1. Arshdeep Bahga and Vijay Madisetti “Internet of Things: A Hands-on Approach”, Universities Press, 2015.
2. Rajkumar Buyya and Amir Vahid Dastjerdi “Internet of Things: Principles and Paradigms”, Morgan Kaufmann; 1st Edition, May 25, 2016.

REFERENCE BOOKS:

1. Matt Richardson & Shawn Wallace “Getting Started with Raspberry Pi”, O’Reilly (SPD), 2014, ISBN: 9789350239759.
2. Waltenegus Dargie, Christian Poellabauer, “Fundamentals of Wireless Sensor Networks: Theory and Practice”.
3. Beginning Sensor networks with Arduino and Raspberry Pi – Charles Bell, A press, and 2013.



Source: https://www.electronicshub.org/wireless-sensor-networks-wsn/

22CS955 WIRELESS SENSOR NETWORKS

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Computer networks.

COURSE DESCRIPTION AND OBJECTIVES:

In this course we will provide an introduction to Wireless Sensor Networks (WSN) and cover latest topics in WSNs. The goal of this course is to give an overview of fundamental problems in the area of WSNs. We will discuss existing solutions for some of these problems. Data aggregation, information dissemination, security issues, power management, localization, topology control, routing, and security, are the topics will be covered in this course.

MODULE-1

UNIT-1 **10L+0T+6P=16 Hours**

INTRODUCTION

Introduction to Wireless Sensor Networks: Background of Sensor Network, Motivations, Performance metrics, Design factors, Sensor node hardware's and software's.

WSN Architecture: Traditional layered stack, roles and challenges, Enabling technologies in WSN, Applications of WSN, Physical layer and transceiver design considerations in WSNs.

UNIT-2 **14L+0T+10P=24 Hours**

MAC PROTOCOLS FOR WSN

Medium Access Control Protocols for WSN: Introduction, Fundamentals of MAC Protocols, Contention-Free, Contention-Based and Hybrid MAC Protocols, Data aggregation and fusion, Distributed data bases.

Localization: Global location (GPS-based) and relative location (Beacon-based). Localization methods: anchor-free, anchor-based, range-free, range-based. Clustering in WSN, Types of clustering.

PRACTICES:

- Implement different network topologies in WSN using NS2 Simulator/Arduino boards.
- Implement traffic signaling using Arduino boards.
- Establish communication between the two motes with Wi-Fi, XBee, modules on arduino and raspberry pi.
- Create cluster formation with m number of motes in WSN by using Arduino and raspberry pi.
- Collect the sensor geographical location information using Raspberry pi.

MODULE-2

UNIT-1 **10L+0T+6P=16 Hours**

ROUTING PROTOCOLS

Routing protocols for WSN: Introduction, Routing Challenges and Design Issues in WSN, Flooding and its variants, LEACH, Location-based protocols and energy-aware routing.

Transport Control Protocols for WSN: Feasibility of Using TCP or UDP for WSNs, TCP Design Issues, Existing TCPs in WSN: CODA, ESRT, RMST, PSFQ, GARUDA, ATP, Problems with TCP, Performance of TCP.

UNIT-2 **14L+0T+10P=24 Hours**

SECURITY

Security: Fundamentals, Security challenges in WSN, Security Attacks, Protocols and Mechanisms for Security, IEEE 802.15.4 and ZigBee Security, Sensor Network programming, Node-Centric Programming: nesC Language, TinyGALS, Sensor Network Simulators: Network Simulator Tools and Environments.

PRACTICES:

- Implement transmission between mobile nodes based on TCP and CBR traffic in WSN nodes using NS2 simulator.
- Implement a Low Energy Adaptive Hierarchy protocol using Simulation Tool.
- Implement different attack and its preventions in WSN using Arduino boards.

COURSE OUTCOMES:

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

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Design a wireless sensor network for given sensor data using microcontroller, transceiver, middle-ware and operating system.	Create	1	3
2	Evaluate the performance of schedule based and random Medium Access Control protocols for power consumption, fairness, channel utilization and control packet overhead.	Analyze	1	1, 2, 3
3	Evaluate the performance of low energy and geo-graphic routing protocols for power consumption, scalability and latency parameters.	Analyze	2	1, 2, 3
4	Implement solutions to real world problems using various sensors and arduino boards.	Create	2	3

TEXT BOOKS:

1. Dargie, Waltenegus, and Christian Poellabauer. Fundamentals of wireless sensor networks: theory and practice. John Wiley & Sons, 2010.
2. Kazem Sohraby, Daniel Minoli and Taieb Znati, “Wireless Sensor Networks Technology, Protocols, and Applications”, John Wiley & Sons, 2007.

REFERENCE BOOKS:

1. Holger Karl and Andreas Willig, “Protocols and Architectures for Wireless Sensor Networks”, John Wiley & Sons, Ltd, 2005.
2. Zhao and L. Guibas, “Wireless Sensor Networks”, Morgan Kaufmann, San Francisco, 2004.
3. C. S. Raghavendra, K.M.Shivalingam and T.Znati, “Wireless Sensor Networks”, Springer, New York, 2004.
4. Anna Hac, “Wireless Sensor Network Designs”, John Wiley & Sons, 2004.

SKILLS:

- ✓ *To Know the fundamentals of wireless sensor networks and its application to critical real time scenarios.*
- ✓ *To study the various protocols at various layers and its differences with traditional protocols.*
- ✓ *To know the issues pertaining to sensor networks and the challenges involved in managing a sensor network.*