R22 SYLLABUS

DATA SCIENCE

B.Tech. COMPUTER SCIENCE AND ENGINEERING - DATA SCIENCE
TABLE OF CONTENTS

III YEAR I SEMESTER		
22TP301	Soft Skills Laboratory	
22CS206	Design and Analysis of Algorithms	
22CS303	Web Technologies	
22DS301	Introduction to Data Analytics	
III YEAR II SEMESTER		
22TP302	Quantitative Aptitude and Logical Reasoning	
22AM202	Digital Logic and Computer Organization	
22CS204	Computer Networks	
22CS207	Operating Systems	
IV YEAR I SEMESTER		
22DS401	IoT and Cloud Analytics	
22DS402	Text Analytics and Natural Language Processing	
22CS402	Big Data Analytics	
IV YEAR II SEMESTER		
22DS403	Project Work	
DEPARTMENT ELECTIVES		
22DS801	Data Wrangling	
22DS802	Speech Processing and Analytics	
22DS803	Social, Web and Mobile Analytics	
22DS804	Time series analysis and Forecasting	
22DS805	Visual Analytics	
22CS801	Advanced Data Structures	
22CS803	Advanced JAVA Programming	
22CS806	Deep Learning	
22CS811	Mobile Application Development	
22CS815	Optimization Techniques	
22AM802	Compiler Design	
22AM803	Computer Vision	
22AM806	Data Warehousing and Data Mining	
22AM807	Evolutionary Computing	
22AM808	Nature Inspired Computing Methods	
22AM809	Soft Computing	
22AM810	Introduction to Software Engineering	
22AM811	Kernel Methods for Pattern Analysis	
Data Science – Minor Courses		
22DS901	Big Data Analytics	
22DS902	Data Science using Python	
22DS903	Introduction to Python Programming	
22DS904	Statistical Methods and Data Visualization	
22DS905	Machine Learning	
22CS909	Capstone Project	

FOREWORD

The vital aspect of CSE Data Science (CSE-DS) program is to offer fundamental skills required for developing business solutions that are optimised through data analysis. In the recent years, there is an escalating demand for professionals having data science skill set in the industry. CSE-DS program assures data science roles for graduates by making them proficient with the tools and technologies that are helpful for gathering, analysing, and interpreting business data.

R22 Curriculum enables CSE-DS students to gain the essentials of applied statistics, applied mathematics, and computer science. Further, the courses on business data establish the context and the need for the data, while the machine learning and analytics related courses equip the students to use and analyse this data towards impactful conclusions. The hands-on experience the students acquire throughout the program makes them industry ready

Salient Features of R22 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-DS 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.V.B.Subramanyam, Professor, Department of CSE, NIT Warangal.
3. Dr.B.Venkata Ramana, Assoc. Professor & HoD, Department of CSE, IIT Tirupathi.
4. Dr.V. Radha, Assoc. Professor, IDRBT, Hyderabad.
5. Dr.Nagesh Bhattu Sristy, Asst. Professor, Department of CSE, NIT AP.
6. Dr.M.Dinesh, Research Scientist, Philips, Bangalore.

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. Venkatesulu Dondeti,
Head, Depat. 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 AND ENGINEERING - DATA SCIENCE

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

- **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.**
- **Nurture research and consultancy activities amongst students and faculty by providing State-of-art facilities and Industry-Institute Interaction.**
- **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 - Data Science

Program Educational Objectives (PEOs)

Program Educational Objectives (PEOs) are established through a consultation process. PEOs are broad statements that describe the career and professional accomplishments that the graduates should achieve within three to five years after their graduation.

Graduates of the UG-CSE program will be able to

PEO1: Pursue career in electrical and allied fields in private/ public sector (or) as an entrepreneur.

PEO2: Design, invent and develop novel technology and find creative, innovative solutions to engineering problems through interdisciplinary approach.

PEO3: Apply professional knowledge to solve technical and social problems in economical way by following ethics.

Program Specific Outcomes (PSOs)

The students will be able to –

PSO1: **Professional Competence:** Ability to apply the core concepts of statistics, mathematics and data science while accomplishing objectives of business organizations by leveraging various software tools.

PSO2: **Research, Entrepreneurial and Social Competence:** Ability to design and develop futuristic models suitable for solving local and societal problems of varying complexities.

Program Outcomes (POs)

Program Outcomes (POs), are attributes acquired by the student at the time of graduation. The POs given in below, ensure that the POs are aligned to the Graduate Attributes (GAs) specified by National Board of Accreditation (NBA).

These attributes are measured at the time of Graduation.

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

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

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

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

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

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

PO7: **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

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

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

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

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	2	0	2	3
22EE101	Basics of Electrical and Electronics Engineering	2	0	2	3
22CS103	IT Workshop and Tools	0	2	2	2
22TP105	Problem Solving through Programming – I	2	2	2	4
22EN102	English Proficiency and Communication Skills	0	0	2	1
22SA101	Physical Fitness, Sports and Games – I	0	0	3	1
22TP101	Constitution of India	0	2	0	1
Total		**9**	**8**	**13**	**19**
		30 Hrs			

I Year II Semester

Course Code	Course Title	L	T	P	C
22MT105	Advanced Engineering Mathematics	3	2	0	4
22MT107	Discrete Mathematical Structures	2	2	0	3
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
22CS104	Python Programming	2	0	2	3
22SA103	Physical Fitness, Sports and Games – II	0	0	3	1
22SA102	Orientation Session	0	0	6	3
Total		**12**	**6**	**17**	**23**
		35 Hrs			

Department Subject is extension of Basic sciences

COURSE STRUCTURE - R22

R22 B.Tech. 4 YEAR DEGREE PROGRAMME

II Year I Semester

Course Code	Course Title	L	T	P	C
22ST206	Probability and Statistical Analysis	3	2	0	4
22TP201	Data Structures	2	2	2	4
22MS201	Management Science	2	2	0	3
22DS201	Data Handling and Visualization	2	0	2	3
22CS201	Database Management Systems	2	2	2	4
22DS202	Digital Logic and Computer Orginization	2	0	2	3
22CT201	Environmental Studies	1	1	0	1
22SA201	Life Skills - I	0	0	2	1
	Total	14	7	12	23
	NCC / NSS / SAC / E-cell / Student Mentoring/ Social activities/ Publication	0	0	0	1
	Total	**14**	**7**	**12**	**24**
		33 Hrs			

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
22AM205	Object Oriented Programming	2	0	4	4
22CY205	Algorithms and Analysis	2	2	2	4
22DS204	Airtificial Intelligence	3	0	2	4
22DS202	Advanced Statistical Methods	2	0	2	3
22SA202	Life Skills - II	0	0	2	1
	Open Elective – 1	3	0	0	3
	Total	**12**	**2**	**16**	**21**
	Minor / Honours - 1	3	0	2	4
		15	**2**	**18**	**25**
	Total	**35 Hrs**			

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
22CS303	Web Technologies	2	0	4	4
22AM204	Machine Learning	3	0	2	4
22CS207	Operating Systems	2	0	2	3
22DS301	Inter-Disciplinary Project – Phase I	0	0	2	0
22DS303	Industry Interface Course	1	0	0	1
	Department Elective – 1	3	0	2	4
	Open Elective – 2	3	0	0	3
	Total	**14**	**0**	**14**	**20**
	NCC/ NSS/ SAC/ E-cell/ Student Mentoring/ Social activities/ Publication with good impact factor (Only 2 students can claim 1 paper/patent). These credits maybe earned on or before the end of VI semester	0	0	0	1
	Minor / Honours - 2	3	0	2	4
	Total	**16**	**0**	**16**	**25**
		32 Hrs			

III Year II Semester

Course Code	Course Title	L	T	P	C
22TP302	Quantitative Aptitude and Logical Reasoning	0	4	0	2
22CY204	Computer Networks	3	0	2	4
22DS303	Data Analytics	2	0	2	3
22DS203	Formal Languages and Automata Theory	2	2	0	3
22DS304	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**	**4**	**8**	**21**
	Minor / Honours - 3	3	0	2	4
	Total	**17**	**4**	**10**	**25**
		31 Hrs			

COURSE STRUCTURE - R22

R22 B.Tech. 4 YEAR DEGREE PROGRAMME

IV Year I Semester

Course Code	Course Title	L	T	P	C
22DS403	Cloud Computing and Analytics	3	0	2	4
22AM301	Deep Learning	3	0	2	4
22CS402	Big Data Analytics	3	0	2	4
	Department Elective – 3	3	0	2	4
	Department Elective – 4	3	0	2	4
	Total	**15**	**0**	**10**	**20**
	Minor / Honours – 4	3	0	2	4
	Total	**18**	**0**	**12**	**24**
		30 Hrs			

IV Year II Semester

Course Code	Course Title	L	T	P	C
22DS402/ 22DS403	Internship/Project Work	0	2	22	12
	Total	**0**	**2**	**22**	**12**
	Minor / Honours – 5 (for project)	0	2	6	**4**
	Total	0	4	28	**16**
		32			**16**

for interaction between Guide and students

R22 B.Tech. 4 YEAR DEGREE PROGRAMME

COURSE STRUCTURE - R22

Department Electives

Course Code	Course Title	L	T	P	C
22DS801	Data Wrangling	2	0	4	4
22DS802	Speech Processing and Analytics	3	0	2	4
22DS803	Social, Web and Mobile Analytics	3	0	2	4
22DS804	Time series analysis and Forecasting	3	0	2	4
22DS805	Visual Analytics	3	0	2	4
22CS801	Advanced Data Structures	2	2	2	4
22CS802	Advanced JAVA Programming	2	2	2	4
22CS806	Deep Learning	3	0	2	4
22CS811	Mobile Application Development	2	0	4	4
22CS815	Optimization Techniques	3	2	0	4
22AM802	Compiler Design	3	2	0	4
22AM803	Computer Vision	3	0	2	4
22AM806	Data Warehousing and Data Mining	3	0	2	4
22AM807	Evolutionary Computing	3	2	0	4
22AM808	Nature Inspired Computing Methods	3	2	0	4
22AM809	Soft Computing	3	2	0	4
22AM810	Introduction to Software Engineering	3	0	2	4
22AM811	Kernel Methods for Pattern Analysis	3	0	2	4

Honours for CSE-DS

Course Code	Course Title	L	T	P	C
	Student can opt any four courses from department electives which he/she did not pursue under department elective courses.	3	0	2	4
		3	0	2	4
		3	0	2	4
		3	0	2	4
	Capstone Project / Any other Course	0	2	6	4

Minor Course

Course Code	Course Title	L	T	P	C
22DS901	Big Data Analytics	3	0	2	4
22DS902	Data Science using Python	2	2	2	4
22DS903	Introduction to Python Programming	2	2	2	4
22DS904	Statistical Methods and Data Visualization	3	0	2	4
22DS905	Machine Learning	3	0	2	4
22CS909	Capstone Project	0	2	6	4
	Total	**15**	**10**	**0**	**20**

II YEAR

COMPUTER SCIENCE AND ENGINEERING - DATA SCIENCE

B.Tech.

I SEMESTER

22ST206	-	Probability and Statistical Analysis
22TP201	-	Data Structures
22MS201	-	Management Science
22CS201	-	Database Management Systems
22DS201	-	Data Handling and Visualization
22AM202	-	Digital Logic and Computer Orginization
22CT201	-	Environmental Studies
22SA201	-	Life Skills - I

II SEMESTER

22TP203	-	Advanced Coding Competency
22TP204	-	Professional Communication
22AM205	-	Object Oriented Programming
22CY205	-	Algorithms and Analysis
22DS204	-	Airtificial Intelligence
22DS202	-	Advanced Statistical Methods
22SA202	-	Life Skills - II

COURSE CONTENTS

I SEM & II SEM

22ST206 PROBABILITY AND STATISTICAL ANALYSIS

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Basic mathematics.

COURSE DESCRIPTION AND OBJECTIVES:

This course is an introduction to statistical methods used to analyse the engineering problems. The main objective of the course is to impart the knowledge about probability, random variables, statistical distributions with design of experiments and these methods helps the students to analyse the practical situations through engineering applications.

MODULE-1

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

PROBABILITY:

Introduction, definition (classical and axiomatic approach), addition theorem (extension to n event also), Boole's inequality, conditional probability, multiplication theorem (for independent event also), Bayes theorem.

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

RANDOM VARIABLES:

Random variable, discrete random variable- probability mass function, discrete distribution function, continuous variable- probability distribution function, two-dimensional random variable- joint probability distribution function, marginal and conditional probability function.

PRACTICES:

- Applications of addition theorem.
- Applications of multiplication theorem.
- Applications of conditional probability.
- Computation of probability for univariate random variable.
- Computation of probability for two-dimensional random variable.

MODULE-2

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

DISCRETE PROBABILITY DISTRIBUTIONS:

Discrete probability distributions: Bernoulli distribution, moments of Bernoulli distribution, Binomial Distribution-Applications and properties of BD, Poisson distribution- Poisson Distribution-Applications and properties of PD.

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

CONTINUOUS DISTRIBUTIONS AND DESIGN OF EXPERIMENTS:

Continuous Probability Distributions: Normal distribution, properties, characteristics, applications of ND, Uniform distribution, exponential Distribution-Applications and properties of exponential distribution.

Design of Experiments: completely randomized design, Randomized complete block design. Chi square tests-goodness of fit and independence of attributes.

SKILLS:

- ✓ *Determination of probability and distribution.*
- ✓ *Generation of probability mass functions.*
- ✓ *Measuring the chance of uncertainty of random variables.*

PRACTICES:

- Applications of discrete probability distributions.
- Identification of one-way and two-ways classification of ANOVA.
- Study of statistical significance difference on more than two groups.
- Study the properties of statistical distributions.
- Designing the statistical plan for various independent groups.

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 probability and statistical methods to discrete and continuous random variables.	Apply	1	1,2,3
2	Measure the chance of random nature of events and calculate how the events will occurs.	Apply	1	1,2
3	Select appropriate discrete or continuous probability functions to calculate probability of events.	Analyze	1,2	1,2,3
4	Design the plan of statistical data to test the significance among the groups.	Create	2	1,2,3,4

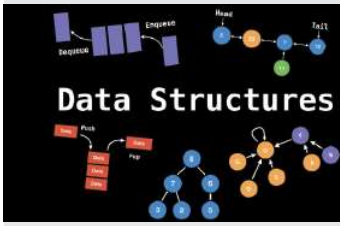
TEXT BOOKS:

1. I.R. Miller, J.E. Freund and R. Johnson, “Probability and Statistics for Engineers”, 9th Edition, Pearson, 2018.
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. Statistics for Business and Economics, Anderson, Dennis J. Sweeney, Thomas A. Williams, 11/e, Cengage Learning, 2011.
3. John Schiller, R. Alu Srinivasan, Murray Spiegel, “Schaum’s Outline of Probability and Statistics” 3rd Edition, The McGraw-Hill Companies Inc, 2009.

22TP201 DATA STRUCTURES



Source: https://www.youtube.com/watch?v=Qmt0QwzEmh0

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Programming in C.

COURSE DESCRIPTION & OBJECTIVES:

This course is aimed at offering fundamentals concepts of data structures and explains how to implement them. It begins with the basic concepts of data, data structures and then introduces the primitive and non-primitive data structures in detail. It forms the basis for understanding various ways of representing data and its usage in different computing applications.

MODULE-1

UNIT-1 **5L+6T+6P = 17 Hours**

DATA STRUCTURES BASICS:

Basic Terminology – data, information, datatype; Data Structures – Introduction, storage structures-sequential and linked storage representations; classification of data structures; Applications of data structures.

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort.

Searching: Linear Search and Binary Search.

UNIT-2 **11L+10T+10P = 31 Hours**

LINKED LISTS AND STACKS, QUEUES:

Linked List: Introduction, Types of linked list – Singly linked list, doubly linked list and circular linked list, representation of linked list, Operations of linked list: Traverse forward/ reverse order, searching, insertion and deletion; Applications of linked lists.

Stack – Introduction, array and linked representations, implementation and their applications; Queue – Introduction, array and linked representations, implementation; Types – Linear, circular and doubly ended queues – operations; Applications of Queues.

PRACTICES:

Problems on Recursion – Level 1

- Find the product of 2 numbers using recursion.
- Find the sum of natural numbers using recursion.
- Find the factorial of a number using recursion.
- Find the Nth term of Fibonacci series using recursion.
- Calculate the power using recursion.
- Write a recursive program for checking if a given number is a prime number.
- Given two integers write a function to sum the numbers without using any arithmetic operators.
- Convert a decimal to binary using recursion.
- Print all factors using recursion.
- Find the maximum product of digits among numbers less than or equal to N.

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.

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.*

- 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 of management

COURSE DESCRIPTION AND OBJECTIVES:

The goal of this course is to analyze the importance of management, significance of operation management and carry out production operations through work-study. Students will be able to analyse the markets, customers, competitors, and then plan HR function effectively. These management practices, functional areas of the organisation will helps the students to build up their career in the corporate world.

MODULE-1

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

INTRODUCTION TO MANAGEMENT:

Concepts of Management and organization- nature, importance and Functions of Management, Systems approach to Management - Taylor's Scientific Management Theory, Fayol's Principles of Management, Mayo's Hawthorne Experiments, Maslow's Theory of Human Needs, Douglas McGregor's Theory X and Theory Y, Herzberg's Two-Factor Theory of Motivation, Leadership Styles, Social responsibilities of Management.

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

OPERATIONS MANAGEMENT:

Principles and Types of Plant Layout-Methods of production (Job, batch and Mass Production), Work Study -Basic procedure involved in Method Study and Work Measurement, Objectives, Need for Inventory control, EOQ, ABC Analysis, Purchase Procedure, Stores Management and Stores Records. Statistical Quality Control: control charts for variables and attributes (simple problems).

PRACTICES:

- Collect some examples with videos for types of production.
- Carry out production operations through work-study.
- Practice problems with Inventory control methods and Quality Control charts.

MODULE-2

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

HUMAN RESOURCES MANAGEMENT:

Concepts of Human Resource Management, Basic functions of HR Manager: Manpower planning, Recruitment, Selection, Training and Development, Placement, Wage and Salary Administration, Promotion, Transfer, Separation, Performance Appraisal, Grievance Handling and Welfare Administration, Job Evaluation, and Merit Rating.

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

MARKETING MANAGEMENT:

Evolution of Marketing, Functions of Marketing Selling Vs Marketing, 4 P's of Marketing – Product Mix - Product Life Cycle – Place Mix – Channels of Distribution – Price Mix – Pricing Methods – Promotion Mix – Tools of Promotions.

PRACTICES:

- Select any Designation in an organization and try to describe its job description and job specifications
- How do you deal with grievances at your work
- Analyze marketing mix in various situations

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	Carry out production operations through work-study	Apply	1, 2	1, 2, 3, 5
2	Analyze the nature and importance of management	Analyze	1	1,2,4,6
3	Significance of Operations Management.	Analyze	1, 2	1,2,5
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. Rajan Saxena: Marketing Management, 4th Edition, TMH, 2013.
2. Dilip Kumar Battacharya, Principles of Management, Pearson, 2012.

REFERENCE BOOKS:

1. Philip Kotler, Kevin Lane Keller, Abraham Koshy and Mithleshwar Jha: Marketing Management, 13th Edition, Pearson Education, 2012.
2. Dipak Kumar Bhattacharyya, Production and Operations Management, Universities Press, 2012.
3. Gary Dessler, “Human Resource Management”, 12th Edition, Pearson- 2012.
4. K.Aswathappa, “Human Resource Management”, Text and Cases”, TMH, 2011.
5. Harold Koontz, Heinz Weihrich, A.R. Aryasri, Principles of Management, TMH, 2010.

Source: https://www.youtube.com/watch?v=IDpB9zF8LBw

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 **10L+6T+4P=20 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 **6L+10T+12P=28 Hours**

RELATIONA 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.

MODULE-2

UNIT-1 **8L+8T+12P=28 Hours**

NORMALIZATION:

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

UNIT-2 **8L+8T+4P=20 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; TheARIES 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 completion of the course, the student will be able to achieve the following outcomes

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”,7thedition,PearsonEducation,2016.
2. Raghu Rama Krishnan and Johannes Gehrke, “Database Management Systems”,3rd edition, TataMcGraw Hill,2013.

REFERENCE BOOKS:

1. Abraham Silberschatz, Henry F.Korth and S.Sudarshan, “Database SystemConcepts”,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.

22DS201 DATA HANDLING AND VISUALIZATION

Source: https://www.analyticsvidhya.com/blog/2021/06/must-known-data-visualization-techniq for-data-science/

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Python programming

COURSE DESCRIPTION AND OBJECTIVES:

This course covers the basics of how large data sets are managed to extract meaningful information. This course imparts knowledge required to understand subtle patterns, trends and correlations necessary to understand the data. In addition, this course covers skills required to leverage data and to reveal valuable insights using real example data sets drawn from a variety of different disciplines. Effective data visualization is an important tool as it allows us to quickly examine large amounts of data, expose trends efficiently, exchange ideas with key players, and influence decisions. This course allows the students to work with various tools for visualization of data from a variety of fields.

MODULE-1

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

DATA STRUCTURES:

NumPy: Creating Arrays, Arrays Operations, Multidimensional Arrays, Arrays transformation, Array Concatenation, Array Math Operations, Multidimensional Array and its Operations, Vector and Matrix operations.

Pandas Data structures - Series and data fames, working with 1D and 2D data- Creation, accessing, manipulation, various operations

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

SUB SETTING, FILTERING, AND GROUPING:

Sub setting the Data Frame, The unique Function, Conditional Selection and Boolean Filtering, Setting and Resetting the Index, The Group By Method, Aggregating

Detecting Outliers and Handling Missing Values: Outlier detection, Missing Values in Pandas, Filling and dropping missing Values in Pandas, Outlier Detection and removing duplicates.

PRACTICES:

- Write code to perform the following operations on Numpy arrays:
  i. Create a 2D Numpy array with 24 elements of size 4x6 and retrieve the last three rows, retrieve the first two column values, retrieve the sum of the second row, retrieve the sum of first column, and display the max value index in the array
  ii. Create a 2D Numpy array with 42 elements of size 7x6, add a new row, Delete an existing column, replace a specific value, and identify how many values are less than given x
  iii. Create a 1D-array with 64 elements, Reshape the array into 4, 2x8 arrays, also reshape the array into other possible shapes, Convert the data type into float, Split the array into three sub-arrays of same size
  iv. Create a 2D Numpy array with 35 elements of size 7x5, identify unique values in the array, identify the existence of duplicates, perform conditional replace operations, insert NaNs, replace NaNs,

SKILLS:

- ✓ *Exploratory Data Analysis*
- ✓ *Data Visualization*
- ✓ *Matplotlib, Seaborn, and Tableau tools usage*

Create the following 2D array using Numpy and perform below operations:

2	3	4	5	6
10	11	12	13	14
18	19	20	21	22
26	27	28	29	30
31	33	34	35	36
37	38	39	40	41

i. Write the code routine to print the masked (gray) colored sub-array
ii. Print the maximum of the fifth row.
iii. Reshape the array (change columns to rows, rows to columns)
iv. Extract all the odd number using conditional logic
v. Find the column wise mean, std and variance

- Apply the following operations on the given csv file
  i. Load data from CSV files
  ii. Retrieve first 10, last 10 rows, 3rd Column and a subgroup
  iii. Query and index operations on the above data frame
  iv. Insert, delete and update your data
  v. Apply aggregate operations
  vi. Apply various filters on the data
  vii. Group, merge, and aggregate data in the data frames
- Apply the following operations on the given csv file
  i. Load the csv and convert to data Frame
  ii. Identify the total number of missing values
  iii. Replace the missing values with a constant, with the Mean of that column, with the mode of that column
  iv. Remove missing values on the original csv file
  v. Apply fill options and replace

MODULE-2

UNIT – 1 **8L + 0T + 8P = 16 Hours**

DATA VISUALIZATION:

Elements of data visualization, Exploration plots: Scatter plots, Line plots, bar plots, box plots, Error-plots, histograms, Kernel-density-estimation plots, Cumulative frequencies, Error-bars, box-plots, bubble-plot, grouped bar charts, pie charts, Advanced plots: correlation, regression, waffle charts, word clouds, Bi-variate, and multivariate plots

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

DATA VISUALIZATION WITH TABLEAU:

Intro to Tableau, Getting started with Tableau Desktop, connecting to the tutorial dataset, Creating the first charts, Filtering and sorting data, creating common visualizations (bar charts, line charts etc.); Advanced visualizations- Creating more advanced chart types, using multiple source tables; Data Storytelling-Intro to data storytelling, Creating a data story in Tableau.

PRACTICES:

- Plotting with matplotlib and Seaborn-Load the given csv file and visualize the data with the help of the following graphical representations:
  a. Line plots b. Bar plots c. Error Plots d. Scatter plots

e. KDE Plots f. Heat Maps g. Box Plots h. Pie graph

i. Histogram j. multiple graphs in single figure k. saving figures

- TABLEAU for visualization

  o Analyze the given patients data and, based on their information, predict and infer the risk of their health. Then integrate all this analysis into Tableau for easy consumption of the end-users.

  o Analyze the data of sales of a company and infer the past sales numbers of a company and then forecast their sales for the coming quarters and years.

  o Analyze the dataset of marketing campaigns and visualize the performance of various marketing campaigns.

  o Analyze a dataset of product-related information, analyze the trends and showcase the availability of any product at any given point in time.

  o Analyze a dataset of flight-related information, consider different factors of a flight and infer accurate trends for flight prices and visualize.

COURSE OUTCOMES:

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

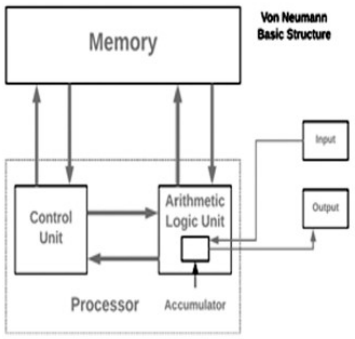
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Application of data pre-process techniques	Apply	1, 2	1
2	Analyze the given data using exploratory and visualization techniques	Analyze	1,2	2
3	Transforming and developing data suitable for Machine learning applications	Design	2	3
4	Tool usage for data handling and visualization	Apply	1,2	1, 5

TEXT BOOKS:

1. Thomas Haslwanter, “An introduction to statistics with python-with applications in the life sciences”, spinger publisher, 2015.
2. Joshua N. Milligan, “Learning Tableau 2020: Create effective data visualizations, build interactive visual analytics, and transform your organization”, 4th Edition, 2020.

REFERENCE BOOKS:

1. Wes McKinney, “Python for data analysis”, 1st Edition, O’Reilly Media, 2012.
2. Joel Grus,”Data Science from Scratch”, O’Reilly Media Inc., 2015.
3. Edward Tufte, “Tableau style guide”.



Source: computer organization & architecture and digital logic design course - Bing images

22AM202 DIGITAL LOGIC AND COMPUTER ORGANIZATION

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Algebra.

COURSE DESCRIPTION AND OBJECTIVES:

The emphasis is on understanding the design of computer and its components. This course introduces the basic knowledge on number systems, design of combinational and sequential circuits. The course mainly focuses on micro operations, memory organizations and input output organization.

MODULE-1

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

NUMBER SYSTEMS AND BOOLEAN ALGEBRA:

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

Boolean Algebra: Fundamental concepts of Boolean algebra basic theorems and propertiesof Boolean Functions.

Gate-Level Minimization: Canonical and standard forms - SOP and POS forms, Digital Logic gates, The map method – two, three, four variable K map; POS and SOP simplification; Don't care conditions;

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

COMBINATIONAL LOGIC CIRCUITS:

Combinational Logic Circuits: combinational circuits analysis, design procedure; Half adder, Full adder, Half subtractor, Full subtractor, Binary adder/subtractor; Binary Incrementor, Decimal Adder, Binary multiplier; Magnitude comparator; Decoders; Encoders; Multiplexers; De-Multiplexer.Synchronous Sequential Logic: Sequential circuits; storage elements - latches, flip flops; Flipflop conversion; counters.

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 greater 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 multiplexers.
- Design a 5-to-32line decoder with four 3-to-8-line decoders with enable and one 2-to-4-line decoder.
- 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, 01 to 11, 11 to 10, 10 back to 00 and repeats until the clock pulse is active.
- Design a 4-bit binary synchronous counter using T flip-flop.

MODULE-2

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

INTRODUCTION:

Introduction to Computer Organization and Architecture: Organization and architecture, structure and function; RTL - Register transfer language, register transfer, bus and memory transfers.

Micro Operations: Arithmetic micro operations; Logic micro operations; Shift micro operations; Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes; Computer registers; Computer instructions; Instruction cycle; Memory reference instructions; Register reference instructions, Input/ Output-Interrupt.

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

MEMORY AND INPUT-OUTPUT ORGANIZATION:

Memory Organization: Memory hierarchy; Main memory; Associative memory; Cache memory; Virtual memory.

Input- Output Organization: Asynchronous data transfer; Modes of transfer; Priority interrupt; Direct memory access.

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.
- Design a 4- bit combinational decrementer circuit 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.
- 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.
- 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.

SKILLS:

- ✓ *Digital circuit Design.*
- ✓ *Number system Conversion.*
- ✓ *Analyse the principles of computer architecture.*
- ✓ *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*

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 and Boolean algebra rules & Karnaugh map method for reducing the Boolean functions.	Apply	1	1
2	Analyse Combinational and sequential digital circuits for the given problem statement.	Analyse	1	2
3	Design of Combinational and sequential digital circuits for the given problem statement and improve the performance by reducing the complexities.	Creating	1	3
4	Demonstrate computer architecture concepts related to the design of modern processor, memory, and I/Osystems.	Analyse	2	1, 2
5	Evaluate the performance of a processor and memory in terms of speed, size and cost.	Evaluation	2	1, 2

TEXT BOOKS:

1. M Morris Mano and Michael D. Ciletti, “Digital Design”, 5th edition, Pearson Education, 2013.
2. M.Moris Mano, “Computer Systems Architecture”, 3rd edition, Pearson/Prentice Hall India, 2007.

REFERENCE BOOKS:

1. John F.Wakerly, “Digital Design Principles and Practices”, Third 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.
4. Carl Hamacher, ZvonkoVranesic and SafwatZaky, “Computer Organization”, 5th edition, Tata McGraw Hill, 2007.
5. William Stallings, “Computer Organization and Architecture”, 6th edition, Pearson/Prentice Hall India, 2007.

22CT201 ENVIRONMENTAL STUDIES

Image source: Biogas plant at VFSTR

Hours Per Week :

L	T	P	C
1	1	0	1

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 **4L+4T+0P=8 Hours**

INTRODUCTION TO ENVIRONMENT: NATURAL RESOURCES, ECOSYSTEMS AND BIODIVERSITY:

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

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

BIODIVERSITY AND CONSERVATION:

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

PRACTICES:

- Visit to a Biogas plant, Solar Power plant.
- Visit to a local area: 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 **4L+4T+0P=8 Hours**

ENVIRONMENTAL POLLUTION AND CLIMATE CHANGE:

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

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

POLLUTION CONTROL DEVICES AND WASTEWATER TREATMENT TECHNOLOGIES:

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

SKILLS:

- ✓ *Create a bio-diversity 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

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

Hours Per Week :

L	T	P	C
0	0	2	1

PREREQUISITE KNOWLEDGE: Programming in C, Data Structures.

COURSE DESCRIPTION AND OBJECTIVES:

This course helps to understand the impact of the choice of data structures and design strategies to solve the problem in an efficient manner. This course also provides the understanding of advanced graph applications and also throw light in tractable intractable problems.

MODULE-1

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

STACKS, QUEUES AND SINGLE LINKED LISTS:

PRACTICES:

Problems On Stacks & Queues

- Check if given stack of integers are consecutive or not (could be ascending or descending).
- Find the maximum sum in a sliding window using queues.
- Given a queue of integers, rearrange the elements by interleaving the first half with the second half.
- Given an integer k and a queue of integers, reverse the order of the first k elements of the queue.
- Given a maze in the form of a rectangular matrix filled with O, X or M where O represents an open cell, X represents a blocked cell and M represents landmines, find the shortest distance of every open cell in the maze from its nearest mine.
- For a given parenthesis expression, check whether it is balanced parenthesis or not.
- Reverse a number using stack.
- You are given a string s consisting of lowercase English letters. A duplicate removal consists of choosing two adjacent and equal letters and removing them. We repeatedly make duplicate removals on s until we no longer can.
- Find first Unique character in a string (Queue).
- Implement Tower of Hanoi problem.

Problems On Linked Lists

- Given a random pointer to a random node in a singly linked list, clone the list.
- Given a list rotate the list to the right by k places.
- Remove duplicates from a sorted list.
- Find fractional node in a singly linked list.
- Sort a linked list using constant space complexity.
- Delete a node in start, middle, end of Singly linked list.
- Add a node in start, middle, end of Singly linked list.
- Find whether given single linked list is circular or not.
- Arrange a singly linked list in Descending order.
- Addition of two numbers using Singly Linked List.

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

DOUBLY LINKED LISTS, CIRCULAR LINKED LISTS:

PRACTICES:

Problems on Double Linked Lists and Circular Linked Lists

- Implement a clockwise rotation of a doubly linked list by N places.
- Count triplets in a sorted doubly linked list whose product is equal to a given value x.
- Find the product of all prime nodes in a doubly linked list.
- Find the count of common nodes in two doubly linked lists.
- Find pairs with given product in a sorted doubly linked list.
- Delete all the even nodes of a circular singly linked list.
- Count nodes in a circular linked list.
- Delete all prime nodes from a circular singly linked list.
- Exchange first and last nodes in a circular linked list.
- Reverse a doubly circular linked list.
- Linear search using a stack of incomplete sub problems.
- 1 2 3 4 5 6 in stack S is push X is pop, SSSSXXSSSXXX.
- Recursively remove all adjacent duplicates.
- Check if a given singly linked list is a palindrome using stack.
- Convert a multilevel singly linked list to a singly linked list.
- Remove duplicates from an unsorted doubly linked list.
- Sort a doubly linked list using insertion sort.
- Check if a doubly linked list of characters is palindrome or not.
- Swap Kth node from beginning with Kth node from end in a Double Linked List.
- Convert a Binary Tree into Double Linked List.

MODULE-2

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

TREES:

PRACTICES:

Problems on Trees

- Given a sorted doubly linked list, convert it into a balanced BST.
- Given a singly linked list with data in the ascending order, convert it into a height balanced BST.
- Print the leaf to root path for every leaf node in a binary tree.
- Write a function to implement the reversed level order traversal of a binary tree.
- Truncate a given binary tree to remove nodes that lie on a path having sum less than K.
- Find the vertical sum in a given binary tree.
- Delete minimum & Maximum element from a BST.
- Implement Inorder, preorder and postorder tree traversal techniques.
- Print Kth largest element in a BST.
- Implement Zig-Zag tree traversal.

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.

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.*

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 LABORATORY

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	2 &1	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.*
- ✓ *Understand and practice specific functions and vocabulary in a business context.*
- ✓ *Produce short business reports, proposals and correspondence.*
- ✓ *Write various business documents through reading techniques.*

Source: https://www.datasciencecentral.com/wp-content/uploads/2021/10/866750746 2.jpeg

22AM205 OBJECT- ORIENTED PROGRAMMING

Hours Per Week :

L	T	P	C
2	0	4	4

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 **6L+0T+12P=18 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 **10L+0T+20P=30 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.

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".

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*

MODULE-2

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

1	1	1
1	1	0
1	0	1

➔

2	2	2
2	2	0
2	0	1

  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:

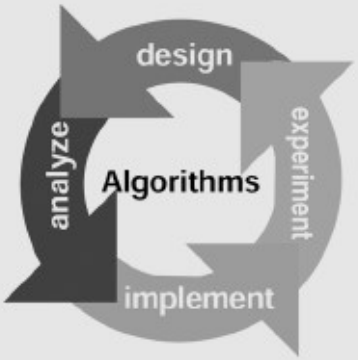
CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Apply object oriented concepts on real time scenarios.	Apply	1	1,2
2	Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes	Apply	1, 2	1,2
3	Design and develop Java applications to solve real 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.



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

22CY205 ALGORITHMS AND ANALYSIS

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.

- 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.

22DS204 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
3	0	2	4

PREREQUISITE KNOWLEDGE: Probability & 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 **12L+0T+8P=20 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 **12L+0T+8P=20 Hours**

PROBLEM SOLVING:

State Space Problem; Searching: Uniform search, InformedSearch: 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 behavior, 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.

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

KNOWLEDGE REPRESENTATION&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, Planninggraphs.

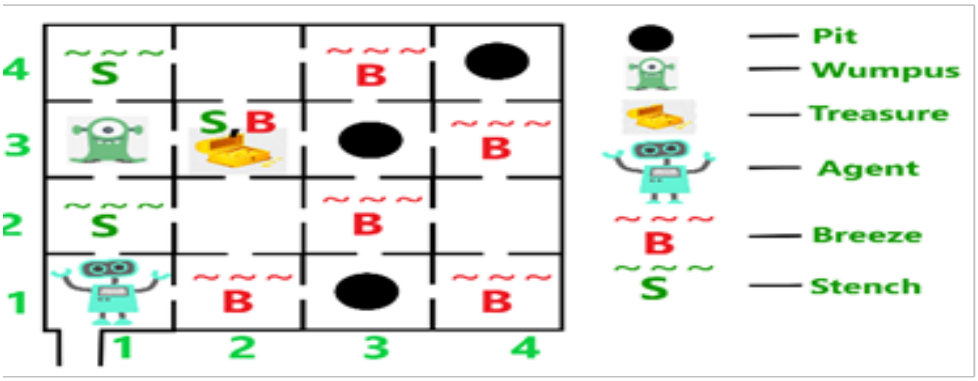
UNIT-2 **12L+0T+8P=20 Hours**

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).
- Write a program to 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 an Artificial Neural Network by implementing the Backpropagation 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 testdata 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	2	2
3	Apply and evaluate the searching strategies to achieve the goal for a given situation.	Apply	1	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”, Fourth Edition, PearsonEducation, 2010.

REFERENCE BOOKS:

1. David Poole, Alan Mackworth, Randy Goebel, ”Computational Intelligence : a logical approach”,Oxford University Press. 2018.
2. G. Luger, “Artificial Intelligence: Structures and Strategies for complex problemsolving”, FourthEdition, Pearson Education, 2010.
3. J. Nilsson, “Artificial Intelligence: A new Synthesis”, Elsevier Publishers, 2013.

22DS202 ADVANCED STATISTICAL METHODS

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Python Programming, Probability theory and Statistical Distributions

COURSE DESCRIPTION AND OBJECTIVES:

. Statistics is critical for data science and this course imparts sufficient knowledge required for data science. Students in this course receive an overview of statistical methods from an experimental design perspective. Students will review statistical sampling, hypothesis, Linear Regression and other related skills. Rather than calculations, the course focuses on interpretation, analysis and communication of the results of statistical analysis. This course makes students familiar with various python libraries that are useful for statistical analysis.

MODULE-1

UNIT-1 **08L+0T+08P=16 Hours**

INTRODUCTION: Need of statistics in data science. Python packages for statistics. Population and samples: Introduction, Sampling techniques, random sampling, clusters sampling, systematic sampling, stratified sampling techniques; Types of data

DESCRIPTIVE STATISTICAL ANALYSIS: Types of Statistical Inference, Descriptive Statistics, Inferential Statistics, Importance of Statistical Inference for data science; Descriptive Statistics- Measures of Central Tendency: Mean, Median, Mode, Mid-range; Measures of Dispersion- Range, Variance, Mean Deviation, Standard Deviation, quartiles and percentiles, Inter Quartile Range; Moments- Skewness and Kurtosis.

UNIT-2 **08L+0T+08P=16 Hours**

ESTIMATION

Introduction, Statistical Inference, Classical Methods of Estimation. Estimating the Mean, Standard Error of a Point Estimate, Prediction Intervals, Tolerance Limits, Estimating the Variance, Estimating a Proportion for single mean, Difference between Two Means, between Two Proportions for Two Samples and Maximum Likelihood Estimation.

PRACTICES:

- Sampling and Resampling:
  - o Generate a population of random numbers.
  - o Generate multiple samples using Random sampling with and without random sampling.
  - o Load a balanced dataset and visualize the class distribution.
  - o Load an imbalanced dataset and visualize the class distribution.
- Interpreting Data Using Descriptive Statistics: Compute Mean, Median, Mode, Standard Deviation, Variance, Co-variance, Interquartile Range and Skewness for two different datasets and write your interpretations about these statistical measures. Which measure is best suitable? Justify.
- Generating Samples from Probability Distributions:
  - o Generate a set of random numbers (which corresponds to a uniform distribution) using the function rand and plot its histogram. What is the shape of this histogram and why?

o Investigate how the shape of the histogram is affected by the number of random **numbers** you have generated.

o Similarly generate numbers using Bernoulli, Binomial distributions and plot a histogram and check the shape.

o Generate numbers using exponential and Poisson distributions and plot a histogram and check the shape

- Under the assumption of binomial distribution, use Maximum likelihood approach and estimate the parameters.
- Under the assumption of Bernoulli distribution, use Maximum likelihood approach and estimate the parameters.

SKILLS:

- ✓ *Exploratory data analysis.*
- ✓ *Statistical data analysis.*
- ✓ *Scikit learn tool usage.*

MODULE-2

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

STATISTICAL DATA ANALYSIS :

Propositional logic: Hypothesis Tests: Typical analysis procedure- Data screening and outliers, Normality check, Hypothesis concept, p-value, Interpretation of p-value, types of Errors, sensitivity and specificity.

HYPOTHESIS TESTS FOR STATISTICAL ANALYSIS: z-test, student t-test, one-way chi-square test, chi-square contingency test, Analysis of variance (ANOVA)- One-way ANOVA and Two-way ANOVA.

UNIT-1 **08L+0T+08P=16 Hours**

CORRELATION AND REGRESSION ANALYSIS

Linear correlation- Correlation coefficient, rank correlation, Pearson Correlation, Correlation Coefficient for Bivariate Frequency Distribution; Regression-General linear regression model, uni-variate regression, Bi-variate regression, Multi-variate regression, regression co-efficients, Co-efficient of Determination, Linear regression analysis with python.

PRACTICES:

- Hypothesis tests: Implement the following three popular statistical techniques for hypothesis testing: Chisquare test, T-test and ANOVA test (Calculate the Test Statistic and P-value by running a Hypothesis test that well suits your data and Make Conclusions).
- Identify Coefficient of correlation between every pair of features in the given dataset and report your inferences
- Using the Gapminder database, We would like to see if an increasing Internet usage results in an increasing suicide rate. A study shows that other factors like unemployment could have a great impact. Accept or reject the hypothesis? Justify your answer
- Linear Regression Analysis: Download house prediction dataset and explore the data, Prepare the dataset for training, Train a linear regression model, and Make predictions and evaluate the model.

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	Application of exploratory and statistical analysis on data	Apply	1, 2	1
2	Analyze data by evaluating various statistical measures	Analyze	1, 2	2
3	Installation and usage of python tools for statistical analysis.	Apply	1, 2	1, 5

TEXT BOOK:

1. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, Probability & Statistics for Engineers & Scientists, 9th Ed. Pearson Publishers, 2012.
2. Thomas Haslwanter, “An introduction to statistics with python-with applications in the life sciences”, spinger publisher, (2015).

REFERENCE BOOKS:

1. T.T. Soong, Fundamentals of Probability And Statistics For Engineers, John Wiley & Sons Ltd, 2004.
2. Sheldon M Ross, Probability and statistics for Engineers and scientists, Academic Press.

III

YEAR

COMPUTER SCIENCE AND ENGINEERING - DATA SCIENCE

B.Tech.

COURSE CONTENTS

I SEM & II SEM

I SEMESTER

22TP301	-	Soft Skills Laboratory
22CS303	-	Web Technologies
22AM204	-	Machine Learning
22CS207	-	Operating Systems
22DS302	-	Inter-Disciplinary Project - Phase-I
22DS303	-	Industry Interface Course

II SEMESTER

22TP302	-	Quantitative Aptitude and Logical Reasoning
22CS204	-	Computer Networks
22DSXXX	-	Data Analytics
22DS203	-	Formal Languages and Automata Theory
22DS304	-	Inter-Disciplinary Project - Phase-II

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, 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.

22CS303 WEB TECHNOLOGIES

Hours Per Week :

L	T	P	C
2	0	4	4

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+16P=24 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+16P=24 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.

- 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+16P=24 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+16P=24 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.

22AM204 MACHINE LEARNING

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Probability & Linear Algebra, Python language.

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 onthe 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.

f. Implement Ridge regression model
g. Implement LASSO regression model
h. Report your observations on the above models for house prediction
e. Observe Variance Inflation Factor(VIF)

- Heart disease prediction:
  a. Implement a logistic regression model to predict whether an individual is suffering fromheart 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 appropriatedata 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 demonstratethe FIND-Salgorithm 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.	Design	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. EthemAlpaydin, “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. AurélienGéron, “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).

22CS207 OPERATING SYSTEMS

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

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.

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.

F**ile 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 thatsuffer 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.

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

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.

22CY204 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 **8L+0T+6P= 14 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 **16L+0T+10P= 26 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

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.*

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.

22DS303 DATA ANALYTICS

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Basic Mathematics, Machine Learning.

COURSE DESCRIPTION AND OBJECTIVES:

A data analytics course is designed to students how to collect, store, organize, and analyze data for various purposes. The course covers topics such as descriptive statistics, graphical methods, probability, hypothesis testing, modelling, data manipulation, and database concepts. The course also provides practical examples and frameworks for managing analytics projects and using data insights to improve decision-making.

The objectives of a data analytics course are to:

- To explore the fundamental concepts of data analytics. Understand data representation formats and techniques and how to use them.
- To understand the various search methods and visualization techniques. Experience with a wide range of data analytics tools including Excel, SQL databases, R, and visualization and reporting software.
- To apply various machine learning techniques for data analysis. Develop a computational thinking approach to problem-solving and use programs and scripting to solve data tasks.
- Convert data into information via appropriate analysis, represent these results in ways that can be readily communicated to others in the organization, and use the information to make evidence-based business decisions.

MODULE-1

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

INTRODUCTION TO DATA ANALYTICS:

Data Analytics Overview-Importance of Data Analytics-Types of Data Analytics-Descriptive Analytics-Diagnostic Analytics-Predictive Analytics- Prescriptive Analytics-Benefits of Data Analytics.

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

DATA VISUALIZATION FOR DECISION MAKING:

Data Types, Measure of Central Tendency-Measures of Dispersion- Graphical Techniques-Skewness & Kurtosis-Box Plot-Descriptive Stats-Sampling Funnel-Sampling Variation-Central Limit Theorem-Confidence interval.

PRACTICES:

- To perform Exploratory Data Analysis on Automobile data.
- To Perform Linear Regression study on the 3D printing dataset.
- To determine mechanical properties from stress strain curve data.
- To perform Machine Predictive Maintenance.
- To develop Manufacturing Cost Model.

MODULE-2

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

TEXT ANALYTICS:

Natural language Basics-Processing and understanding Text-Text Summarization-Text similarity and Clustering -Text Classification-Semantic and Sentiment analysis.

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

DATA ANALYTICS BUSINESS CASE STUDIES:

Sales Forecasting- Inventory management prediction- Census- Dimensionality Reduction-Predictive Modelling- HR analytics.

PRACTICES:

- To perform Thermal Analysis from IOT Devices temperature readings
- Case Study on Sales Forecasting
- Case Study on Inventory management prediction
- Case Study on dimensionality reduction

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	Recognize the characteristics of data that make it useful to solve real-world problems.	Analyze	1	1, 2, 3
2	Identify and apply appropriate algorithms for analyzing the data for a variety of problems.	Apply	1	1, 2
3	Synthesise insights from a range of analyses to develop evidence-based solutions to complex problems.	Analyze	1, 2	1, 2, 3
4	Identify and apply appropriate algorithms for analyzing the data for a variety of problems in finance, risk, and healthcare.	Evaluate	2	1, 2, 3, 4

TEXT BOOKS:

1. Ravindra Gogineni “Data Analytics”, First Edition, 2020.
2. Data Analytics Made Accessible by Dr. Anil Maheshwari.

REFERENCE BOOKS:

1. P. R. Vittal, “Mathematical Statistics”, Margham Publications, Chennai, 2018.
2. Sandhya Arora and Latesh Malik “ Data Science and Analytics with Python”, Universities Press, 2023.

.

22DS203 FORMAL LANGUAGES AND AUTOMATA THEORY

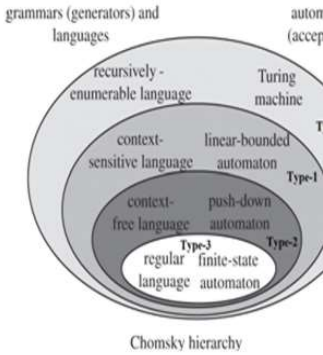
Hours Per Week :

L	T	P	C
2	2	0	3

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.



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

MODULE-1

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

REGULAR EXPRESSION:

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;

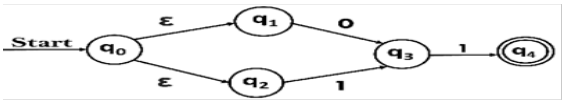
PRACTICES:

- 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,andaacd: 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}

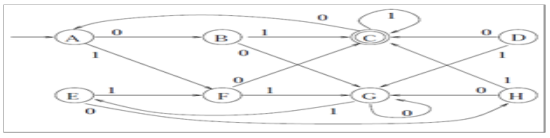
SKILLS:

- ✓ *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={ap | p is a prime number}
  (b) L={bn | n=i2 and i>1}
  (c) L={ WWR | W is (a,b)*}
  (d) L={anbn+1 |n≥1}

MODULE-2

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

PUSH DOWN AUTOMATA (PDA):

PUSH DOWN 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.

PRACTICES:

- Construct CFG for the following
  a) L={anbn|n>1}
  b) L={WWR | W is (a,b)*}
  c) L={ap | 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
  (a) L={ap | p is a prime number}
  (b) L={bn | n=i2 and i>1}
  (c) L={ WWR | W is (a,b)*}
  (d) L={anbn+1 |n≥1}

- Convert the following language or PDA to CFG
  (a)L={anbn|n≥1} and
  (b) δ(q,0,z)={(q,xz)}
  δ(q,0,x)={(q,xx)}
  δ(q,1,x)={(q,x)}
  δ(q, ,x)={(p,)}
  δ(p,1,x)={(p,xx)}
  δ(p, ,x)={(p,)}
  δ(p, 1,z)={(p,)}
- Construct PDA for the following languages
  a) L={0n1m | n≥m}
  b) L={anbn|n≥1}
  c) L= $\{w \mid w \in \{a,b\}^*$
  d) L={w| na(w) >nb(w)}
  e) L={0n12n | n>0}
  f) L = Where wRis reverse of w
  g) L= Where wRis reverse of w
- Construct PDA for the following languages
  a) L= {anbncn | n >1}
  b) L={anbmambn | n,m≥1}
- Find the CFG corresponding to PDA whose transition mapping is as follows:
- Construct Turing Machine for the following languages
  a) L={anbn | n ≥ 1}
  b) L={02n1n |n >0 }
  c) L={wwr | w is (0+1)*}
  d) L={anbncn|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.	Apply	1	1, 3
2	Design different finite state machines to perform various operations.	Apply, Analyze, Design	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, Design	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.

IV

YEAR

COMPUTER SCIENCE AND ENGINEERING - DATA SCIENCE

B.Tech.

I SEMESTER

- 22DS401 - Cloud Computing and Analytics
- 22AM301 - Deep Learning
- 22CS402 - Big Data Analytics

II SEMESTER

- 22DS402 - Internship
  22DS403 - Project Work

COURSE CONTENTS

I SEM & II SEM

22DS401 CLOUD COMPUTING AND ANALYTICS

Source: https://www.ibm.com/blogs/internet-of-things/edge-iot-analytics/

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Computer fundamentals, Programming fundamentals.

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. There are billions of devices in homes, industries, cities, hospitals, cars, and thousands of other places. With the rapid increase of devices, you increasingly need solutions to connect them, and collect, store, and analyze device data. Data in its raw form is not always useful. Data need to be processed to transform into information. In this course, you will learn how to collect and analyse sensor data. You will learn data processing, data visualization and machine learning algorithms for predictive analytics.

MODULE-1

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

IOT FUNDAMENTALS

Introduction to Internet of Things; Physical design & Functional Block of IoT. Core IoT 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**

CLOUD COMPUTING FUNDAMENTALS

Cloud Computing Fundamentals: Motivation for Cloud Computing, The Need for Cloud Computing, Defining Cloud Computing, Definition of Cloud computing, Cloud Computing Is a Service, Cloud Computing Is a Platform, Principles of Cloud computing, Five Essential Characteristics, Four Cloud Deployment Models. Tools: AWS, Microsoft Azure.

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.
- Configure Amazon Virtual Private Cloud (VPC)
  a. Create your own VPC.
  b. Create public subnet.
  c. Create private subnet.
  d. Create a Internet gateway and attach to your VPC.
  e. Create Public Routing Table, associate subnet and add routing tables. Create Private Routing Table, associate subnet and add routing tables.

MODULE-2

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

SYSTEMS DESIGN & DEVELOPMENT:

IoT system building blocks, Arduino, Node MCU– Board details, IDE programming; Raspberry Pi-Model and Interfaces.

SKILLS:

- ✓ *Sensor Identification and IoT system design.*
- ✓ *Sensor data analysis & Visualization*
- ✓ *Tool usage for developing IoT applications.*

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

PROGRAMMING AND DATA ANALYTICS:

Embedded C vs Python; Operating systems for constrained devices; Domain Specific IoT Application, Collecting Data from sensors locally, Sending Sensor Data to IoT Cloud (ThingSpeak).

Matlab: Introduction, Data Analysis, Data Visualization, Machine Learning.

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 of controlling LED through switch using Python in Raspberry Pi board.
- Design and Implementation to find obstacles through sensor using Python in Raspberry Pi board.
- ThingSpeak account creation
- Sending Light Sensor Values to IoT Cloud
- Sending Temperature and Humidity Values to IoT Cloud
- Sensor Data Visualization
- Energy savings with Anomaly Detection using Z-Score Analysis
- Correlation between Temperature and Humidity and Regression
- Temperature Prediction using Polynomial Regression.

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 IoT projects for sensor data collection	Design	1	1, 3
2	Design solutions of various IoT applications using Raspberry Pi/Arduino/Node MCU boards.	Design	2	3
3	Apply the fundamentals of machine learning and statistics to extract value from IoT data	Apply	1	1
4	Analyze different business use-cases for IoT data	Analyze	2	2

TEXT BOOKS:

1. Arshdeep Bahga and Vijay Madisetti “Internet of Things: A Hands-on Approach”, Universities Press, 2015, ISBN: 9788173719547.
2. K. Chandrasekhran, “Essentials of cloud Computing” CRC press, 2014.

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”, 2010.
3. Beginning Sensor networks with Arduino and Raspberry Pi – Charles Bell, A press, and 2013.
4. Andrew Minteer “Analytics for the Internet of Things (IoT): Intelligent analytics for your intelligent devices”, Packt Publishing; 1st edition, 24 July 2017.

22AM301 DEEP LEARNING

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= 20Hours**

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:

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

- 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 forBlack 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 completion of the course, the student will be able to achieve the following outcomes:

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	Analyzeperformance 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 Goodfellow and YoshuaBengio and Aaron, “Deep Learning”, 1st Edition, An MITPress Book, 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://miro.medium.com/max/844/0*ARAg3FnAzy2e02Wy.png
max/844/0*ARAg3FnAzy2e02Wy.png

22CS402 BIG DATA AND ANALYTICS

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Basics of Databases, Data mining.

COURSE DESCRIPTION AND OBJECTIVES:

This course serves as an introductory course to gain knowledge on analysing 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 **12L+0T+8P=20 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 **12L+0T+8P=20 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 **12L+0T+8P=20 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.

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

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*

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.
  - o input file format has year, sales of all months and average sales.
  - o Year Jan Feb Mar April May Jun July Aug Sep Oct Nov Dec Average.
- Map Reduce program to find Dept wise salary.
  - o 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.

DEPARTMENT
ELECTIVES

COMPUTER SCIENCE AND ENGINEERING - DATA SCIENCE

B.Tech.

22DS801	-	Data Wrangling
22DS802	-	Speech Processing and Analytics
22DS803	-	Social, Web and Mobile Analytics
22DS804	-	Time series analysis and Forecasting
22CS801	-	Advanced Data Structures
22CS802	-	Advanced JAVA Programming
22CS808	-	Mobile Application Development
22AM802	-	Compiler Design
22AM803	-	Computer Vision
22AM806	-	Data Warehousing and Data Mining
22AM807	-	Evolutionary Computing
22AM808	-	Nature Inspired Computing Methods
22AM809	-	Soft Computing
22AM810	-	Introduction to Software Engineering

COURSE CONTENTS

I SEM & II SEM

22DS801 DATA WRANGLING

Image source: https://www.itworldcanada.com/blog/is-your-ai-data-wrangling-out-of-control/431069

Hours Per Week :

L	T	P	C
2	0	4	4

PREREQUISITE KNOWLEDGE: Python programming.

COURSE DESCRIPTION AND OBJECTIVES:

This course offers several standard steps of the data wrangling process like importing data, tidying data, string processing, HTML parsing, working with dates and times, and text mining.
In a data science project, more likely that the data is available in a file, a database, or extracted from documents such as web pages, tweets, or PDFs. In these cases, the first step is to import the data and process it. This course teaches converting the raw data to the tidy form which is a critical step for any data scientist.

MODULE-1

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

DATA ACQUISITION:

Introduction: Overview of Data Preparation, Data Preparation and Data Wrangling,

Import data into Pandas from various data sources: Fetching stored data- CSV, Excel, Pdf, text, multiple text files, RDBMS (SQL Tables), pickle and JSON; importing data from clipboards, Working with binary data formats, Web scraping-beautiful soup for reading and parsing of web pages, reading data from XML, reading data from an API, Reading Image files using PIL, read multiple files using Glob,

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

DEEP DIVE INTO DATA WRANGLING:

Data Integration-Combining, merging, and joining data sets, string and text processing using regular expressions, Random sampling and random reordering of rows; Data Cleaning- selecting numerical and categorical variables, Cleaning the numerical features, Cleaning the categorical features, KNN blank filling, Column Transformer and make_column_selector; Transforming Numerical features – power transformation, binning, binarization, data transformation based on mappings, Encoding Categorical data- One-hot encoding, Ordinal encoding, Label encoding of the target variable; Scaling- Normalization, Standardization, Robust scaling.

PRACTICES:

- Write code to perform the following operations:
  i. Load the data stored in different files formats: CSV, Excel, txt, ZIP, database, JSON, pickle
  ii. Saving data into different files formats: CSV, Excel, txt, ZIP, database, JSON, pickle
  iii. Web Scraping- Use requests module to retrieve data from any website of your interest, improve readability of the extracted data using BeautifulSoup library
  iv. Reading Image Folders: retrieve images stored in our local folders and visualize few samples from each class
  v. Load a CSV file, change column names, apply required transformations on the data
- Load a CSV file, apply binning to transform numeric data to categorical data
- Load a CSV file, encode the categorical values- transforms non-numerical labels into numerical labels, One-Hot encoding, and Binary encoding
- Merge multiple data frames, split a data frame into multiple data frames, apply different joins on two data frames
- Data pre-processing- formatting data to standardize it and make it consistent, normalizing data, grouping data values into bins

SKILLS:

- ✓ *Data Acquisition, and Engineering*
- ✓ *NLTK toolkit usage*
- ✓ *Creation of New Datasets*

MODULE-2

UNIT–1 **8L+0T+16P= 24 Hours**

LEARNING THE HIDDEN SECRETS:

Advanced list comprehension and zip function, Date and time data types and tools, Time series basics, Time zone handling

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

TEXT MINING USING NLTK:

Installing NLTK, What is text wrangling, Creating and loading a Text Corpus, Sentence Tokenization, Word Tokenization, removing special Characters, Expanding contractions, Removing Stop words, Correcting words-repeated characters, Stemming & lemmatization, part of speech tagging, chunking

PRACTICES:

- Work With Datetime Format in Python - Time Series Data, Convert integer to DateTime, Extract DateTime into Different Columns, Assemble DateTime from Different Columns
- Create Pandas DateTime Series using date_range() method
- Load data that contains dates and times
  - o Convert strings to datetime
  - o Assemble a datetime from multiple columns
  - o Get year, month and day
  - o Get the week of year, the day of week, and leap year
  - o Get the age from the date of birth
  - o Improve performance by setting date column as the index
  - o Select data with a specific year and perform aggregation
  - o Select data with a specific month and a specific day of the month
  - o Select data between two dates
  - o Handle missing values
- Loading text data, apply exploratory data analysis to understand the data, tokenize, convert to lower case, remove stop words, apply stemming and lemmatization and apply pos tagging

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 pre-processing, transformation and cleaning techniques on data acquired from multiple sources	Apply	1, 2	1
2	Analyze the given data and apply required transformations to the data	Analyze	1,2	2
3	Designing datasets for machine learning and data science projects	develop	1,2	3
4	Tool usage for data acquisition and management	Apply	1,2	6

TEXT BOOKS:

1. Python for Data Analysis: Data Wranglingwith Pandas, NumPy, and IPython by Wes McKinney (2017)

REFERENCE BOOKS:

1. Wes McKinney, “Python for data analysis”, 1st Edition, O’Reilly Media, 2012
2. Joel Grus,”Data Science from Scratch”, O’Reilly Media Inc., 2015.

22DS802 SPEECH PROCESSING AND ANALYTICS

Image source: https://www.aiperspectives.com/speech-recognition/ Image file name: Speech Recognition

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Machine Learning.

COURSE DESCRIPTION AND OBJECTIVES:

Oral communication is possibly the most natural, common, and direct mode of human communication. The course provides the foundation knowledge on speech production and perception along with processing of speech signal in digital domain. This course offers an in-depth introduction to automatic speech recognition (ASR), the problem of automatically converting speech into text. This class also covers theoretical and practical aspects of machine learning (ML) techniques that are employed in large-scale ASR systems. Apart from teaching classical algorithms that form the basis of statistical speech recognition, this class will also cover the latest deep learning techniques that have made important advances in achieving state-of-the-art results for speech recognition and related problems in spoken language processing.

MODULE-1

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

SPEECH PROCESSING:

Phonetics –Articulatory Phonetics -Phonological Categories -Acoustic Phonetics and Signals - Speech Synthesis –Text Normalization –Phonetic and Acoustic Analysis -Diphone Waveform synthesis – Evaluation-Automatic Speech Recognition –Architecture -Hidden Markov Model to Speech -MFCC vectors -Acoustic Likelihood Computation -Evaluation. Triphones – Discriminative Training -Modeling Variation. Computational Phonology- Finite-State Phonology –Computational Optimality Theory -Syllabification -Learning Phonology and Morphology.

UNIT-2 **12L+0T+8P= 16 Hours**

SPEECH ANALYSIS:

Features, Feature Extraction and Pattern Comparison Techniques: Speech distortion measures – mathematical and perceptual – Log Spectral Distance, Cepstral Distances, Weighted Cepstral Distances and Filtering, Likelihood Distortions, Spectral Distortion using a Warped Frequency Scale, LPC, PLP and MFCC Coefficients, Time Alignment and Normalization – Dynamic Time Warping, Multiple Time – Alignment Paths.

PRACTICES:

- Sampling Frequency and Bit Resolution for Speech Signal Processing
  - o Write a Scilab program to convert input speech signal sampled at 44.1 kHz to 22.05 kHz. The program also should plot spectra of selected segments both from original and resampled versions. By comparing respective segment spectra, comment on the effect of changing sampling frequency.
  - o Write Scilab program to quantize input speech signal at 16 bit to 4 bit data by keeping the sampling frequency constant. Write down your observation by comparing with 16 bit data
- Short Term Time Domain Processing of Speech:
  - o Write a SCILAB program for computing short term energy using frame sizes of 100,200 & 500 msec and all with a frame shift of one sample. Compare the reults with the case of 30,50 & 100 msec case given in the procedure, Write down the observation using a rectangular window.

SKILLS:

- ✓ *Modelling Speech Systems*
- ✓ *Understanding Acoustic systems*
- ✓ *Tools to process speech signals*

o Modify the above program for the case of hamming window. write down your observation.

o Modify the short term ZCR program by not including the factor "N"(frame length) in the relation. compute the ST ZCR for window size of 30,50 & 100 msec. compare the same with the earlier case given in the procedure section. Do you find any difference? comment.

o Write the pitch estimation program in SCILAB using frame sizes of 10,50 & 100 msec, each with a shift of 10 msec. Compare the nature of plots in three different cases & comment.

- Identification of Voice/Unvoiced/Silence regions of Speech:

o Write a Scilab program that reads a speech file and plots the waveform, spectrum and autocorrelation sequence of any three voiced segments present in the given speech signal.

o Write a Scilab program that reads a speech file and plots the waveform, spectrum and autocorrelation sequence of any three unvoiced segments present in the given speech signal.

o Write a Scilab program that reads a speech file and plots the waveform, spectrum and autocorrelation sequence of any three silence region present in the given speech signal.

o Given a speech signal, How can you tell where one word ends and another word begins? Do you notice any patterns in terms of how different sounds look visually, on either the waveform or the spectrogram?

MODULE-2

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

SPEECH MODELLING AND RECOGNITION:

Speech Modelling: Hidden Markov Models: Markov Processes, HMMs – Evaluation, Optimal State Sequence – Viterbi Search, Baum-Welch Parameter Re-estimation, Implementation issues.

Speech Recognition: Large Vocabulary Continuous Speech Recognition: Architecture of a large vocabulary continuous speech recognition system – acoustics and language models – n-grams, context dependent sub-word units; Applications and present status

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

DEEP LEARNING TECHNIQUES FOR SPEECH RECOGNITION:

Neural Network-Based Acoustic Modelling: RNN-based models, Language modelling, RNN-based language models, End-to-end neural architectures for ASR, Speech Synthesis, Convolutional Neural Networks in Speech.

PRACTICES:

- Acoustic modeling of monophone units: The objective is to experiment with methods for training, recognition and evaluation for designing a simple speech recognition application. The task of the exercise is to perform experiments with a small recognition application (sequence of digits). Assume that the grammar is predefined. The exercise consists of the following sub-tasks:
  - o Enter the phonetic transcriptions of the digits in a pronunciation dictionary file.
  - o Record training and test data.
  - o Train monophone, single-mixture phone models.
  - o Evaluate the recognition performance on the test data and on-line.
  - o Enter the results and discuss them shortly

  The exercise will also give you some familiarity with the Hidden Markov Toolkit (HTK).Generate a population of random numbers.
- speaker identification and verification systems:
  - o Implement SVM based speaker identification system for 50 speakers with NIST data.
  - o Develop an SVM based Speaker verification system for 50 speakers with NIST data
- Implement an end-to-end deep neural network for automatic speech recognition

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	Pre-processing and feature extraction from speech signals	Apply	1	1
2	Apply statistical and deep models for modelling speech data	Apply	2	1
3	Develop speech processing systems	develop	1, 2	3
4	Use tools to analyze speech signals	Analyze	1, 2	2, 5

TEXT BOOKS:

1. Daniel Jurafsky and James H.Martin Speech and Language Processing, Prentice Hall: 2nd edition,2008.
2. Lawrence Rabiner and Bing-Hwang Juang, “Fundamentals of Speech Recognition”, Pearson Education, 2003.

REFERENCE BOOKS:

1. Steven W. Smith, “The Scientist and Engineer’s Guide to Digital Signal Processing”, California Technical Publishing, 1st Edition, 1997.
2. Thomas F Quatieri, “Discrete-Time Speech Signal Processing – Principles and Practice”, Pearson Education, 1st Edition, 2003.
3. Claudio Becchetti and Lucio Prina Ricotti, “Speech Recognition”, John Wiley and Sons, 1999.
4. Ben gold and Nelson Morgan, “Speech and audio signal processing”, processing and perception of speech and music, Wiley- India Edition, 2006 Edition.
5. Frederick Jelinek, “Statistical Methods of Speech Recognition”, MIT Press

Image source: https://netbasequid.com/blog/what-is-social-media-analytics-why-is-it-important/
Image file name: Social Media Analytics

22DS803 SOCIAL, WEB AND MOBILE ANALYTICS

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Knowledge of any Programming language.

COURSE DESCRIPTION AND OBJECTIVES:

Social media analytics is the ability to gather and assess the data gathered from social channels to support business decisions — and measure the performance of actions based on those decisions through social media. In this course the student will be introduced to important topics of web analytics covers related key concepts using R, Google Analytics, and the Flex dashboard. This course introduces to R libraries that enable the students to perform real-time analysis of website, analyse traffic and customers behaviour in real-time and how to use this information to optimize the performance of a website.

MODULE-1

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

INTRODUCTION TO WEB & SOCIAL ANALYTICS:

What is social media? Need of using analytics, social analytics vs web analytics, Types of web analytics, Current analytics platforms, Open-source vs licensed platforms, Google analytics, IBM social media analytics, Choosing right specifications and optimal solution. Relevant Data & its Collection: Data collection strategy, participating with people centric approach, social graph, Choosing focused data sources and social networks, Facebook social APIs, Live recruiting and remote user research, Types of surveys. Welcome to: KPIs/ Metrics: KPI's, Metrics vs KPI's, Creating KPI's, Basic KPI's, Standard vs critical web metrics, KPI's goals and targets, Specific KPIs, Visualizing the social analytics framework, build scorecards & dashboards to track KPIs, measuring macro & micro conversions, Measuring success for non-ecommerce and B2B websites.

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

MANAGE WEB & SOCIAL MEDIA WITH ANALYTICS:

Explore & evaluate – dashboard, Types of charts, sharing insights with dashboards, Sentiments, Evolving topics, Segmentation, Discovery, Social media content creation process, Competitive intelligence analysis, Website traffic analysis, social signals. Future of Social Media Analytics and Monitoring: Tasks of social media analytics and monitoring, Mashups, integrating customer profile data, colliding data sets for big bang ideas, social interactions and web visits, Benefits of social media integration.

PRACTICES:

- Extracting Google Analytics data in R using RGoogleAnalytics:
  a) Installing packages RGoogleAnalytics and their dependent packages.
  b) Obtain the credentials with the Google Analytics API.
  c) Authenticate the Google Analytics data with the R script and access the Google Analytics data.
  d) Obtain the Profile ID/View ID of the Google Analytics profile for which the data extraction has to be done.
  e) Generate the query and hit it to the Google Analytics API

MODULE-2

SKILLS:

- ✓ *Website traffic Analysis*
- ✓ *Optimize performance of a website*

UNIT-1 **12L+0T+8P= 22 Hours**

MOBILE ANALYTICS:

Introduction to Mobile Analytics: Global Internet usage, Analysing the mobile consumer behaviors, Web analytics Vs mobile analytics, Social media analysis can't tell…, Need of mobile analytics, Mobile communication, WAP- wireless application protocol, GGSN, Layered architecture of mobile computing, Mobile cloud computing architecture.

Mobile Customer Experience Management: Customers end-to-end journey, Customer engagement with mobile apps, Mobile app analytics strategies, Metrics for behavior analysis, Real-time In-app analytics, Voice XML (VXML), Mobile IM and messaging apps, Multichannel campaign optimization, LBS applications for the consumer market.

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

MOBILE ANALYTICS APPLICATIONS:

Mobile analytics for content Publishers & Operators: Mobile Handset Analysis, Screen resolutions in terms of page views, visits and visitors, Mobile operator analysis, Sessions, Content categorization, Mobile operator - RDSN lookup, Full referral URL report, Search term performance, Specific visitor behavior.

Mobile Analytics in Email Marketing: The need of mobile email marketing, Features of email marketing tools, Categories of email marketing reports, Email bounce tracking, Unsubscribe reporting, Individual-level email tracking, Cold calling. Data Functionalities: Data functionalities, Mobile web page tagging, Aggregate containers, Pageviews per annum, Usability analytics, Form field analytics, Attention map, Data archiving, Historical trends, IBM end to end mobile analytical solution with tealeaf integration, Target page HTTPS communication.

PRACTICES:

- Twitter mining for the popular coffee brand "Starbucks"
  a) Installing packages twitter and their dependent packages
  b) Obtain the credentials for Twitter with the twitter account
  c) Setting the certification at Twitter
  d) Searching Twitter and performing sentiment analysis for the tweet named Starbucks.
  e) Doing some plots of the obtained
- Website Data Analysis
  a) Plot past 6 months visitors data for a given website
  b) Recognize loyal visitors
  c) Segmentation and categorization of a website's traffic
  d) Analyze the impact of social media on your visitors
  e) Optimize performance of a website

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 web and social media analytics to gain insights into a website's performance	Apply	1	1
2	Harness the power of social media analytics to improve website or business	Analyze	2	1
3	Use the R-markdown language tools to generate easy-to-understand PDF reports and dashboards	Develop	1, 2	2,5
4	Create an effective measurement model for analytics	Create	1, 2	3

TEXT BOOK:

1. Raghav Bali, Dipanjan Sarkar, Tushar Sharma “Learning Social Media Analytics with R”, Packt Publishing, 2017.

REFERENCE BOOKS:

1. Nathan Danneman, Richard Heimann, “Social Media Mining with R”, · 2014.
2. Dr. Bharti Motwani ,“Data Analytics with R”, Wiley Pubishers, 2015.

22DS804 TIME SERIES ANALYSIS AND FORECASTING

Image Source: https://www.xenonstack.com/blog/time-series-analysis

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Introduction to probability theory and statistics, calculus, Linear algebra and Python Programming.

COURSE DESCRIPTION AND OBJECTIVES:

This course will cover the fundamental techniques in analysing time series data. This course provides an understanding of the different forecasting techniques. It discusses different explorative and decomposition models, Box-Jenkins analysis and ARIMA models for deep understanding of the time series analysis and forecasting.

MODULE–1

UNIT–1 **12L+0T+8P = 20 Hours**

INTRODUCTION TO TIME SERIES AND FORCASTING:

Fundamentals Concepts: Purpose, Time Series, the nature and uses of forecasts, some examples of time series, the forecasting process, data for forecasting, Resources for forecasting,

Statistics Background for Forecasting: Introduction, Graphical displays, Numerical description of time series data, use of data transformation and adjustments, general approach to time series modelling and forecasting, Evaluating and monitoring forecasting model performance.

UNIT – 2 **12L+0T+8P = 20 Hours**

REGRESSION ANALYSIS AND FORECASTING

Fundamentals Concepts of Regression: Introduction, least square estimation in linear regression models, statistical inference in linear regression, prediction of new observations, model adequacy checking, variable selection methods in regression, estimating the parameters in time series regression models.

Exponential Smoothing Methods: Introduction, first-order exponential smoothing, modelling time series data, forecasting.

PRACTICES:

- Suppose that you are in charge of capacity planning for a large electric utility. A major part of your job is ensuring that the utility has sufficient generating capacity to meet current and future customer needs. If you do not have enough capacity, you run the risks of brownouts and service interruption. If you have too much capacity, it may cost more to generate electricity.
  a. What forecasts do you need to do your job effectively?
  b. Are these short-range or long-range forecasts?
  c. What data do you need to be able to generate these forecasts?
- Your company designs and manufactures apparel for the North American market. Clothing and apparel is a style good, with a relatively limited life. Items not sold at the end of the season are usually sold through off-season outlet and discount retailers. Items not sold through discounting and off-season merchants are often given to charity or sold abroad.
  a. What forecasts do you need in this business to be successful?
  b. Are these short-range or long-range forecasts?
  c. What data do you need to be able to generate these forecasts?
  d. What are the implications of forecast errors?

SKILLS:

- ✓ *Acquire the fundamental time series analysis techniques.*
- ✓ *Analyse the various time series techniques for specific applications.*
- ✓ *Know the more advanced topics and current research literature of time series and forecasting.*
- ✓ *Ability to work in industry or in academic research in the field of time series and forecasting.*

- Consider an airline that operates a network of flights that serves 200 cities in the continental United States. What long-range forecasts do the operators of the airline need to be successful? What forecasting problems does this business face on a daily basis? What are the consequences of forecast errors for the airline?
- Apply the method of link relatives to the following data and calculate seasonal indices. Quarterly Figures

Quarter	1995	1996	1997	1998	1999
I	6.0	5.4	6.8	7.2	6.6
II	6.5	7.9	6.5	5.8	7.3
III	7.8	8.4	9.3	7.5	8.0
IV	8.7	7.3	6.4	8.5	7.1

- The following table relates to the to the tourist arrivals during 1990.to 1996 in India:

Years :	1990	191	1992	1993	1994	1995	1996
Tourists arrivals :	18	20	23	25	24	28	30

(in millions)

Fit a straight line trend by the method of least squares and estimates the number of tourists that would arrives in the year 2000.

- Below are given the figures of production (in thousand quintals) of a sugar factory.

Year	Production (thousand quintals)
1993	77
1995	88
1996	94
1997	85
1998	91
1999	98
2002	90

(i) Fit a straight line by the “least squares” method and tabulate the trend values.

(ii) Eliminate the trend. What component of the series are thus left over?

(iii) What is monthly increase in the production of sugar?

- Calculate 5 yearly and 7 yearly moving averages for the following data of the numbers of commercial and industrial failure in a country during 1987 to 2002. Also plot the actual and trend values on a graph.

Year	No.of Failures
1987	23
1988	26
1989	28
1990	32
1991	20
1992	12
1993	12
1994	10
1995	9
1996	13
1997	11

1998	14
1999	12
2000	9
2001	3
2002	1

- Consider the US Treasury Securities rate data in Table B.1 (Appendix B). Find the sample autocorrelation function and the variogram for these data. Is the time series stationary or nonstationary?

  Consider the time series data shown in Table below.

  a. Make a time series plot of the data.

  b. Use simple exponential smoothing with □= 0.2 to smooth the first 40 time periods of this data. How well does this smoothing procedure work?

  c. Make one-step-ahead forecasts of the last 10 observations. Determine the forecast errors.

Period	y_t	Period	y_t	Period	y_t	Period	y_t	Period	y_t
1	48.7	11	49.1	21	45.3	31	50.8	41	47.9
2	45.8	12	46.7	22	43.3	32	46.4	42	49.5
3	46.4	13	47.8	23	44.6	33	52.3	43	44.0
4	46.2	14	45.8	24	47.1	34	50.5	44	53.8
5	44.0	15	45.5	25	53.4	35	53.4	45	52.5
6	53.8	16	49.2	26	44.9	36	53.9	46	52.0
7	47.6	17	54.8	27	50.5	37	52.3	47	50.6
8	47.0	18	44.7	28	48.1	38	53.0	48	48.7
9	47.6	19	51.1	29	45.4	39	48.6	49	51.4
10	51.1	20	47.3	30	51.6	40	52.4	50	47.7

  Reconsider the time series data shown in Table E4.1. a.

  a. Use simple exponential smoothing with the optimum value of □ to smooth the first 40 time periods of this data (you can find the optimum value from Minitab). How well does this smoothing procedure work? Compare the results with those obtained in Exercise 4.1.

  b. Make one-step-ahead forecasts of the last 10 observations. Determine the forecast errors. Compare these forecast errors with those from Exercise 4.1. How much has using the optimum value of the smoothing constant improved the forecasts?

MODULE-2

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

AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA) MODELS:

Introduction, linear models for stationary time series, nonstationary processes, time series model building, forecasting arima processes, seasonal processes.

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

TRANSFER FUNCTIONS AND INTERVENTION MODELS:

Introduction, transfer function models, transfer Function-Noise models, model specification, intervention analysis.

Survey Of Other Forecasting Methods: Multivariate time series models and forecasting, state space models arch and graph models, neural networks and forecasting.

PRACTICES:

- Consider the time series model.

  yt = 150 − 0.5yt−1 + □t

  a. Is this a stationary time series process?

  b. What is the mean of the time series?

  c. If the current observation is y100 = 85, would you expect the next observation to be above or below the mean?

- Consider the time series model

  yt = 50 + 0.8yt−1 − 0.15 + □t

  a. Is this a stationary time series process?

  b. What is the mean of the time series?

  c. If the current observation is y100 = 160, would you expect the next observation to be above or below the mean?

- For each of the ARIMA models shown below, give the forecasting equation that evolves for lead times □ = 1, 2, …, L. In each case, explain the shape of the resulting forecast function over the forecast lead time.

  a. AR(1)

  b. AR(2)

  c. MA(1)

  d. MA(2)

  e. ARMA(1, 1)

  f. IMA(1, 1)

  g. ARIMA(1, 1, 0)

- Consider the transfer function model. Find the forecasts that are generated from this model.

$$y_t = \frac{-0.5 - 0.4B - 0.2B^2}{1 - 0.5B} x_{t-2} + \frac{1}{1 - 0.5B}\varepsilon_t.$$

- Find time series data of interest to you where a transfer function– noise model would be appropriate.

  a. Identify and fit the appropriate transfer function–noise model.

  b. Use an ARIMA model to fit only the yt series.

  c. Compare the forecasting performance of the two models from parts a and b.

- Find a time series of interest to you that you think may be impacted by an outlier. Fit an appropriate ARIMA model to the time series and use either the additive outlier or innovation outlier model to see if the potential outlier is statistically significant.

- Show that an AR(2) model can be represented in state space form and Show that an MA(1) model can be written in state space form.

22CS801 ADVANCED DATA STRUCTURES

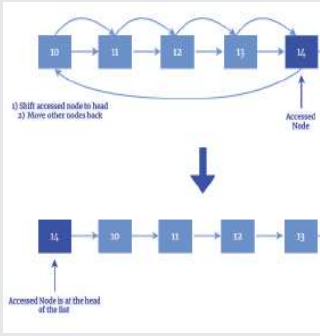


Image Source: https://www.baeldung.com/cs/advanced-data-structures

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**

NUMBER THEORY & COMPUTATIONAL GEOMETRY

Number Theory: Motivation and concept of modulo, Arithmetic operations under modulo, Euclidean algorithm, extended Euclidean algorithm.

Computational Geometry: One Dimensional Range Searching, Two-Dimensional Range Searching.

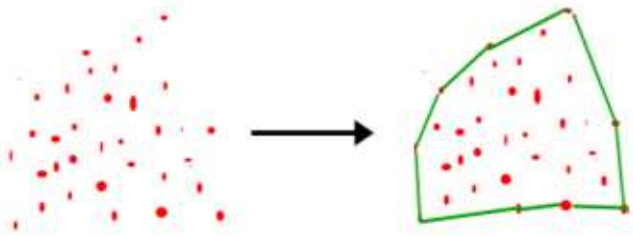
UNIT–2 **8L+8T+8P=24 hours**

TREES:

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

PRACTICES:

- Find GCD of a given number using Euclidean and extended Euclidean algorithms.
- Check whether a given number is prime or not using Sieve of Eratosthenes.
- Implement a program to find the closest pair of points using a divide and conquer strategy. Use the random number generator to generate a large number of points in a unit square as input to the algorithm. Test the correctness of the algorithm by using a brute force method.
- Given a set of points in the plane. the convex hull of the set is the smallest convex polygon that contains all its points of it.



- Implementation of a Red-Black tree operation.
- Implementation of a Splay tree operation.

SKILLS:

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

- 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.

  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**

QUEUES & HEAPS:

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

Heaps: Introduction, types of heaps, heap implementation with an array, priority queue implementation with heaps, heap sort, technique to sort elements, Mergeable heaps, Fibonacci Heaps and operation.

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:

- 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.
- 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	Apply logic and methods behind the major proofs in Number Theory	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. Karl-Dieter Crisman, Number Theory: In context and Interactive by AIM Open Textbook Initiative 7th Edition, 2021.
2. Sahani, Anderson freed, Horowitz, Fundamentals of DATA STRUCTURES in C, Silicon Pr; 2nd Edition, August 1, 2007.

REFERENCE BOOKS:

1. Mark de Berg, Otfried Cheong, Marc van Kreveld, Computational Geometry: Algorithms and Applications:, Springer; 3rd Edition. 2008.
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson, 2004
3. T.Cormen, R.Rivest, C.Stein, C.Leiserson,“Introduction to Algorithms”, , PHI publication, Second Edition, 2004, ISBN 81-203-2141-3.

22CS802 ADVANCED JAVA PROGRAMMING

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Programming.

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 & HIBERNATE:

ORM & Hibernate: 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–I **8L+8T+8P=24 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–II **8L+8T+8P=24 hours**

SPRING & SPRING MVC:

Spring & Spring MVC: 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 frameworks 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.

REFERENCE 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/

22CS808 MOBILE APPLICATION DEVELOPMENT

Image Source: https://www.tatvasoft.com.au/blog/mobile-application-development-methodology/

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=24Hours**

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=24Hours**

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.

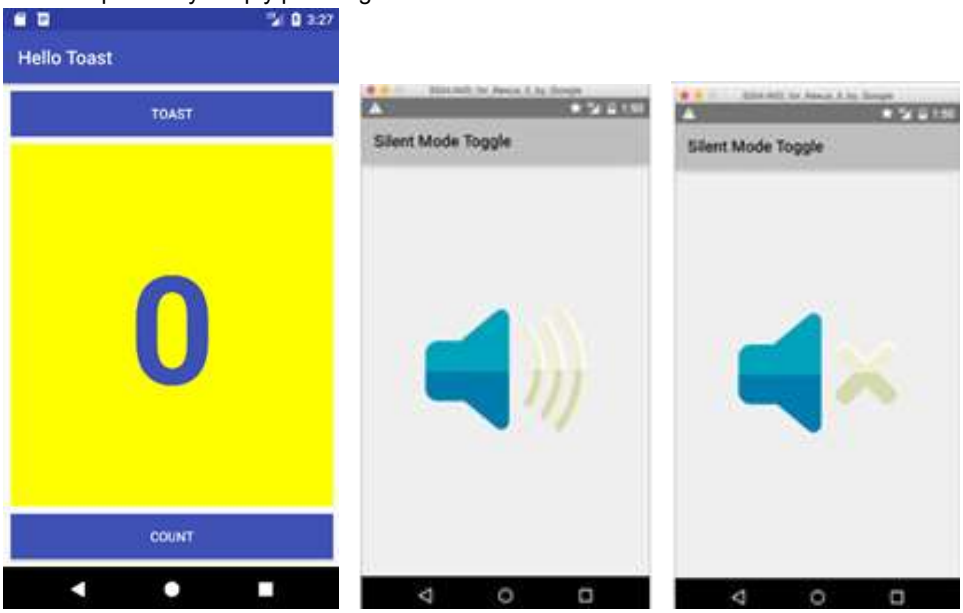
PRACTICES:

- 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 TextView. 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 require-ments.*
- ✓ *Use of suitable advanced components to design mobile apps.*
- ✓ *Utilization of activities, intents, layouts, and views for content.*

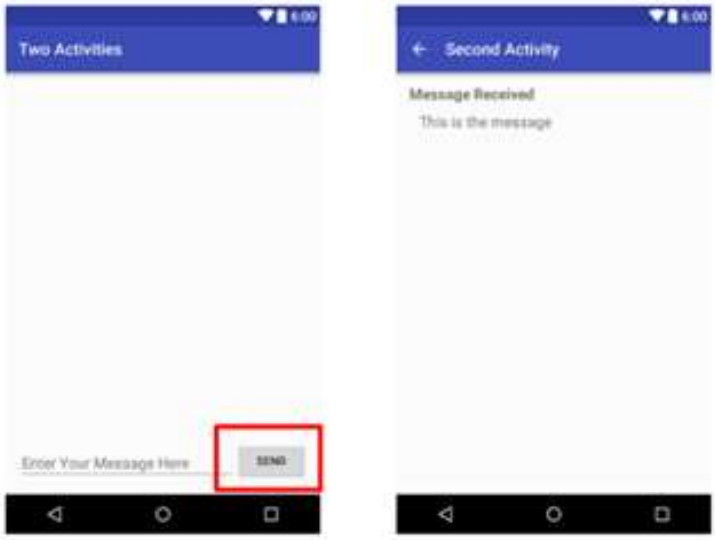
- Design **Silent Model Toggle application:** This app allows the user to toggle the ringer mode on the phone by simply pressing a button.



- In this assignment, students will create and build an app called Two Activities. Students will build the app in four stages.

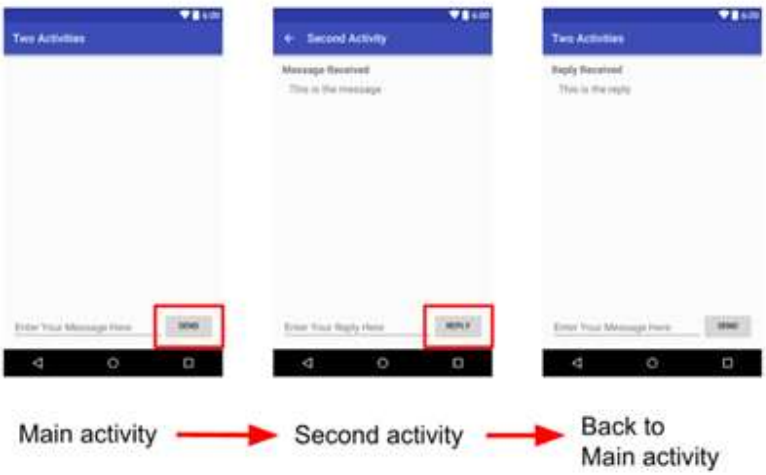
  a. In the first stage, you create an app whose main activity contains one button, Send. When the user clicks this button, your main activity uses an intent to start the second activity.

  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.

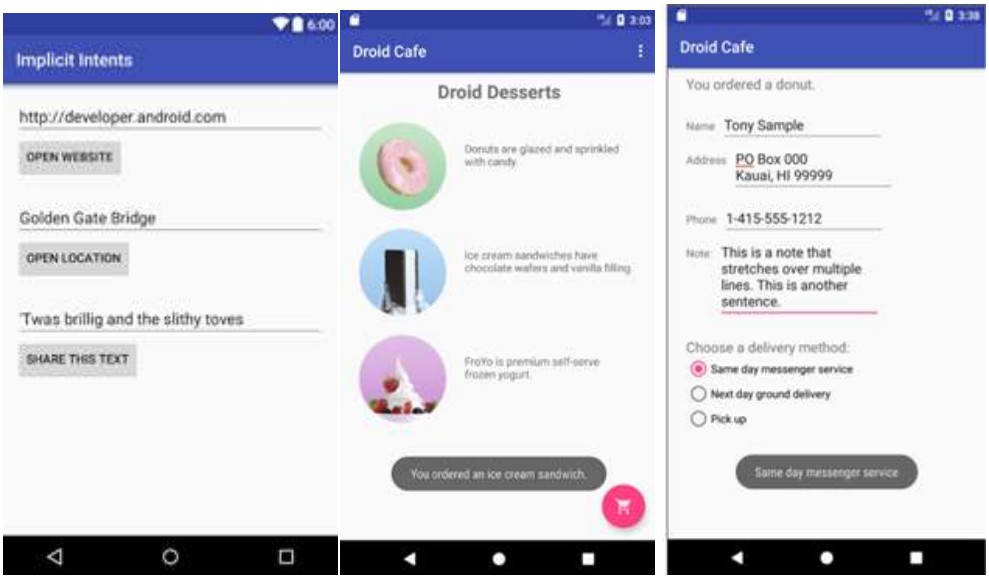
  d. Implement all the Activity lifecycle call back methods to print messages to logcat when those methods are invoked. These log messages will allow you to see when the Activity lifecycle changes state, and how those lifecycle state changes affect your app as it runs.



- 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.

  d. 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.

  e. 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 completion of this course, the students will be able 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. https://aws.amazon.com/mobile/mobile-application-development/
2. https://google-developer-training.github.io/android-developer-fundamentals-course-concepts/.
3. Michael Burton,” Android App Development for Dummies “, 3rd Edition, A Wiley Brand, 2020.
4. Dawn Griffiths & David Griffiths, “Headfirst Android Development A Brain-Friendly Guide” 2nd Edition, O’Reilly, 2015.

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

22AM802 COMPILER DESIGN

Hours Per Week :

L	T	P	C
3	2	0	4

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 analyser, parser, code optimization and code generation.

MODULE-1

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

INTERMEDIATE REPRESENTATIONS:

Three address code; Syntax tree; DAG.

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

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

CODE IMPROVEMENT

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>=0){
  sum = B[0];
  i = 0;
  L1: if (A[i]< B[i]){
  j=i;
  L2:
  if(B[i]>=0){
  sum = sum +B[j];
  }
  j = j+1
  if (j<N) goto L2;
  }
  i = i+1
  if (i<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, McGrawHill, 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.

22AM803 COMPUTER VISION

Image Source: https://www.kdnuggets.com/2020/06/6-easy-steps-implement-computer-vision-application-tensorflow-js.html

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Basic Image Processing, Linear Algebra, Vector Calculus, Python Programming with Open CV and Neuroscience.

COURSE DESCRIPTION AND OBJECTIVES:

This course will cover the fundamentals of Computer Vision. This course provides an understanding of the basic mathematical elements of the image processing and computer vision. It explains and illustrates how the objects are located and identified by the computer in real time environment.

MODULE - 1

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

INTRODUCTION TO COMPUTER VISION:

Basic Concepts Of Image Formation: Introduction and Goals of Computer Vision and Image Processing, Bio-Vision Vs Computer-Vision, Brief history of Image Formation Concepts.

Fundamental Concepts Of Image Formation And It's Transformation: Radiometry, Geometric Transformations, Geometric Camera Models, Camera Calibration, 2D and 3D transformations, 2D and 3D rotations and projections, Image Formation in a Stereo Vision Setup, Image Reconstruction from a Series of Projection

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

IMAGE AND VISION PROCESSING:

Basic Image Processing: Image generation, Image Enhancement, Image Filtering, Colour Image Processing, Basic Image Segmentation techniques.

Image Descriptors and Features: Texture Descriptors, Colour Features, Edges/Boundaries, Object Boundary and Shape Representations, Interest or Corner Point Detectors, Histogram of Oriented Gradients, Scale Invariant Feature Transform, HoG, LBP, Blobs; Scale Space and Scale Selection; Speeded up Robust Features

PRACTICES:

- Explore and show the practical implementation of the analogy between eye and camera and explain about the vital parts for providing the vision in the human eyes and computers?

- Explore and discuss what are the basic image formation techniques and describe with an example how image formation technique got evolved into the present form in terms of the time and space complexities (show the implementations of different algorithms).
- Implement an intelligent and optimized technique using image arithmetic and logical operations to enhance the given input image. Discuss how to choose the best method depending on the input image.
- Explore an intelligent row wise and column wise algorithm for pixel operations to enhance

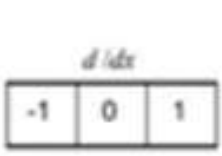
SKILLS:

- ✓ *Acquire the basic image formation knowledge and fundamental image processing techniques.*
- ✓ *Analyse the various Segmentation techniques for specific applications.*
- ✓ *Know more advanced topics and current research literature of Image Processing and Computer Vision.*
- ✓ *Ability to work in industry or in academic research in the field of Computer Vision and Image Processing.*

(highlight the horizontal and vertical edge operations) the given input image.

- Compute the Harris matrix for 3x3 window highlighted below and compare it with 2x2 and 4x4 windows in terms of complexity?

0	0	1	4	9
1	0	5	7	11
1	4	9	12	16
3	8	11	14	16
8	10	15	16	20



- Discuss and implement a robust technique forperforming gray level slicing on medical imageswith and without background noise.
- Investigate how to revise the technique of histogram equalization to enhance the given input image with salt and pepper noise.
- Write a function to compute the Euclidean distance between the original RGB pixel values and the quantized values. Your function should be called compute Quantization Error, should take in inputs origImg, quantizedImg, and should return an output error, where origImg and quantizedImg are both RGB images, and error is a real number.

MODULE - 2

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

VISUAL MATCHING:

Feature: Bag-of-words, VLAD, RANSAC, Hough transform, Pyramid Matching, Stereo vision, Correspondence problem, Motion and Optical Flow.

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

REAL TIME APPLICATIONS OF COMPUTER VISION:

Artificial Neural Network for Pattern Classification, Gesture Recognition, Motion Estimation, Object detection and Tracking, Face Recognition, Instance recognition, Category recognition, and context and scope understanding.

PRACTICES:

- Discuss and compare different image compression techniques in terms of complexity, and propose an intelligent and fast compression technique appropriate to the input image type.
- Given an input image with noise and degradation write an intelligent algorithm to restore the image.
- Your headlights have a radiant intensity of 60 Wsr−1. Determine the irradiance on a sign 2 meters away.
- Investigate different existing image (Indoor and outdoor images) segmentation algorithms. Design your own segmentation algorithm on Vadlamudi region (extract image form google map) and compare with the state of the are methods.
- Explore different existing image (Indoor and outdoor images) segmentation algorithms. Design your own segmentation algorithm and compare with the state of the are methods.
- Given an RGB image, perform clustering in the 3-dimensional RGB space, and map each pixel in the input image to its nearest center. That is, replace the RGB value at each pixel with its nearest cluster's average RGB value. Show the output image for different value of K. Since these average RGB values may not be integers, you should round them to the nearest integer (1 through 255). Your function should be called quantizeRGB, should take in inputs origImg and k, and return outputs outputImg, meanColors, clusterIds. The variables origImg and outputImg

are RGB images, k specifies the number of colors to quantize to, and meanColorsis a Kx3 array of the K centers (one value for each cluster and each color channel). clusterIdsis a numpixelsx1 matrix (with numpixels = numrows * numcolumns) that says which cluster each pixel belongs to.

- A function called detectEdges which takes in as input im, threshold and returns output edges. This function computes edges in an image. im is the input color image, and threshold is a user-set threshold for detecting edges. edges is an Nx4 matrix containing 4 numbers for each of N detected edge points: the x location of the point, the y location of the point, the gradient magnitude at the point, and the gradient orientation (non-quantized) at the point.

  • In this function, first convert the image to grayscale. Then simply compute the gradient magnitude and orientation at each pixel, and only return those (x, y) locations with magnitude that is higher than the threshold. You can reuse code from HW2.

  • At the end, display, save, and include in your submission the thresholded edge image for an image of your choice.

  • Remember that the x direction corresponds to columns and the y direction corresponds to rows.

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	Explorethe fundamental understanding of computer vision and its techniques with its advanced libraries.	Apply	1	1, 2, 12
2	Implement the techniques to have hands on experience with all the techniques.	Apply	1	1, 2, 5, 12
3	Able to design new changes in methods, and make use of different existing techniques to enhance the vision better.	Apply	1	1, 2, 3, 5, 12
4	Capable to analyse and comparereal time models for computer vision problems.	Analyze	2	1, 2, 3, 12
5	Going in depth to find out research outcomes.	Analyze	2	1, 3, 5, 2

TEXT BOOKS:

1. M.K. Bhuyan, “Computer Vision and Image Processing: Fundamentals and Applications”, CRC Press, 2020.
2. Forsyth & Ponce, “Computer Vision-A Modern Approach”, 2nd edition, Pearson Education, 2012.

REFERENCE BOOKS:

1. R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”, 2nd edition, Cambridge University Press, 2004.
2. Simon Prince, “Computer Vision: Models, Learning, and Inference”, Cambridge University Press, 2012
3. Richard Szeliski, “Computer Vision- Algorithms & Applications”, Springer, 2010.

22AM806 DATA WAREHOUSING AND DATA MINING

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Database Management System, Probability and Statistics, Python Programming.

COURSE DESCRIPTION AND OBJECTIVES:

This course introduces the basic concepts, principles, methods, implementation techniques, Data warehousing and 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 **12L+0T+8P=20 Hours**

INTRODUCTION:

Data Warehouse: What is Data Warehouse, A Multidimensional Data Model: From tables to data cubes, Stars, snowflake, and fact constellations (schemas for multidimensional databases), Examples for defining star, snowflake, and fact constellation schemas,OLAP operations in the multidimensional data model.

Data Mining: 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 aretargeted? Major issues in data mining; Data objects and attribute types; Basic statistical descriptions of data;Measuring Data Similarity and Dissimilarity.

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, Histogram Analysis, Concept Hierarchy Generation for Nominal Data.

UNIT-2 **12L+0T+8P=20 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, Constraint based Frequent Pattern Mining.

PRACTICES:

- Creation of multidimensional data model and implementation of various schemas using multidimensional data model.
- Implement various OLAP operations on multidimensional data modelsuch as SLICE,DICE,ROLLUP,DRILLDOWN and PIVOT.
- 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 usingApriori algorithm.
- Extract the interesting association rules from a given dataset using Frequent Pattern growth algorithm.

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.*

MODULE - 2

UNIT-1 **12L+0T+8P=20 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; Other classification methods – KNN; genetic algorithms; Fuzzy algorithm.

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

CLUSTER ANALYSIS:

Partition methods - K means and K medoid; Hierarchical methods;

Agglomerative and divisive method; Density based methods – DBSCAN,OPTICS; Optics; Grid based methods-STING,CLIQUE; 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 and analyze data warehousing concepts and operations	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 BOOK:

1. Jure Leskovec, Anand Raja raman and Jeffrey D Ullman, “Mining of Massive Datasets”, 5th edition, Stanford University, 2014.

22AM806 EVOLUTIONARY COMPUTING

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Design and Analysis of Algorithms.

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 **12L+8T+0P=20 Hours**

INTRODUCTION:

Overview of Philosophy, Nature to Nature Computing, self-Organization, Complexity, Emergence, Bottom-up Vs Top-down approach, determination, chaos and fractals.

Genetic Algorithms: Introduction, Representation of Individuals, Mutation, Recombination, Population Models, Survivor Selection.

Evolution Strategies: Representation, Self-Adaptation, the Ackley Function, Subjective Evolution of Color Mixes.

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

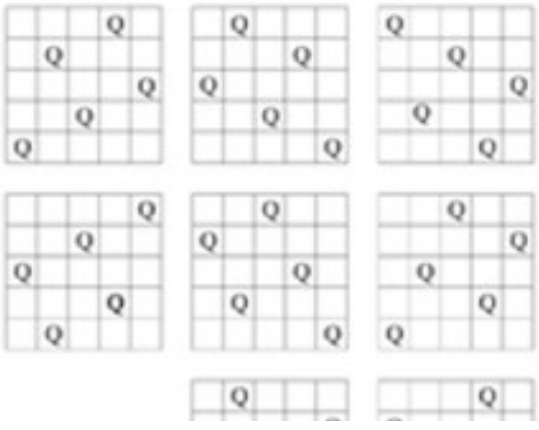
GENETIC PROGRAMMING:

Initialization, Bloat in Genetic Programming, Problems Involving “Physical” Environments, Example Application: Symbolic Regression.

Learning Classifier Systems: 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:

- Explain the usage of fitness function in genetic algorithm with example.
- Solve the five queen’s problems using Genetic algorithm.



SKILLS:

- ✓ *Skills to generate Frameworks of the ideas like Computational. Biology and Computational Psychology*
- ✓ *Construct biological principles and modeled on natural processese valuate and apply*

- Let A ε $R^{m \times n}$ and B ε $R^{p \times n}$. Consider the constrained least squaresproblem:
  minimize ||b – Ax||
  subject to Bx = d
  Show that the problem may not have any solution if rank(B) < p.
- Use the state-wise regression model to prove that the distance between two valid codes equals the minimum number of small variations to obtain a valid code from another valid code

MODULE - 2

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

PARAMETER CONTROL IN EVOLUTIONARY ALGORITHMS:

Introduction: Examples of Changing Parameters, Classification of Control Techniques, Examples of Varying EA Parameters.

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

MULTIMODAL PROBLEMS AND SPATIAL DISTRIBUTION:

Introduction: Multimodal Problems and the Need for Diversity, Implicit Measures, Explicit Diversity Maintenance, Multi objective Evolutionary Algorithms, Example Application: Distributed Co-evolution of Job Shop, Introduction to Memetic algorithms.

PRACTICES:

- How the global and local aspect of memetic algorithm strategy is is used to solve TSP problem?
- How does MOEA help in solving multi-objective optimization problem?

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 various evolutionary computation techniques.	Understand	1	1, 2
2	Analyze real world problems and identify evolutionary algorithms suitable for solving them.	Analyze	1	1, 2, 3
3	Apply techniques for optimization, learning, and design.	Apply	1	1, 2, 3
4	Implement evolutionary algorithms in real world complex problems.	Implement	2	11, 12

TEXT BOOK:

1. A.E. Eiben, J.E Smith, “Introduction to Evolutionary Computing”, 2nd edition, Springer, 2015.

REFERENCE BOOKS:

1. Deb, K, “Optimization for Engineering Design”, PHI India, 2000.
2. Deb, K. “Multi-objective Optimization using Evolutionary Algorithms”, Wiley, Uk, 2001.
3. Melanie Mitchell, “An Introduction to Genetic Algorithms”, MIT Press, 1996.
4. John Koza, “Genetic Programming”, MIT Press, 1992.
5. David Fogel, “Evolutionary Computation”, The Fossil Record, IEEE Press, 1998.
6. Goldberg D. E, “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley, Massachusetts, 1989.

22AM807 NATURE INSPIRED COMPUTING METHODS

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Data Structures and Algorithms.

COURSE DESCRIPTION AND OBJECTIVES:

This course covers the fundamentals of nature inspired techniques which influence computing, study the Swarm Intelligence and Immuno computing techniques and familiarize the DNA Computing.

MODULE-1

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

INTRODUCTION, COMPUTING INSPIRED BY NATURE:

From Nature-to-Nature Computing, Philosophy, Three Branches: A Brief Overview, Individuals, Entities and agents - Parallelism and Distributivity Interactivity.

Computing Inspired by Nature: Evolutionary Computing, Hill Climbing and Simulated Annealing, Darwin's Dangerous Idea, Genetics Principles, Standard Evolutionary Algorithm -Genetic Algorithms.

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

SWARM INTELLIGENCE:

Introduction - Ant Colonies, Ant Foraging Behaviour, Ant Colony Optimization, SACO and scope of ACO algorithms, Ant Colony Algorithm (ACA), Swarm Robotics, Foraging for food, Social Adaptation of Knowledge, Particle Swarm Optimization (PSO)

PRACTICES:

- How would you implement a GA for a two variable numeric function? What would change: the representation, evaluation function, or genetic operators?
- Write a pseudocode for the simple ACO (S-ACO) algorithm considering pheromone evaporation, implement it computationally, and apply it to solve the TSP instance presented. Discuss the results obtained.
- Apply the AS-TSP algorithm to solve the simple TSP instance. Compare the performance of the AS-TSP with that of the genetic algorithm applied to this same problem. Study and discuss the influence of parameters α and β in the performance of the algorithm.

MODULE-2

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

IMMUNO COMPUTING:

Introduction: Immune System, Physiology and main components, Pattern Recognition and Binding, Immune Network Theory- Danger Theory, Evaluation Interaction Immune Algorithms, Introduction – Genetic algorithms, Bone Marrow Models, Forest's Algorithm, Artificial Immune Networks.

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

COMPUTING WITH NEW NATURAL MATERIALS DNA Computing, DNA Molecule, Adleman's experiment, Test tube programming language, Universal DNA Computers, PAM Model, Splicing Systems, Lipton's Solution to SAT Problem, Scope of DNA Computing, From Classical to DNA Computing.

SKILLS:

- ✓ *Describe the needs for present and future computing paradigm changes with examples found in the real world*
- ✓ *Able to develop solutions to problems, accompanied by in-depth analysis and insight*
- ✓ *Demonstrates a competent level of knowledge of NIC concepts, principles, and*
- ✓ *Applications*
- ✓ *Design suitable NIC methods to solve problems and model systems in the real-world contexts*

PRACTICES:

- Implement the clonal selection algorithm CLONALG and apply it to the pattern recognition
- Write a test tube program to verify if a given tube t contains DNA strands with the following sequence of bases: ACTGG.
- A clique Ki is the complete graph on i vertices. Given a graph G = (V,E), determine the largest i such that Ki is a subgraph of G. This problem is known as the maximum clique problem (MCP). Write a test tube program to solve it, and discuss its functioning.

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 fundamental concepts of NP-hardness and computational complexity	Apply	1	1, 2, 12
2	Understand the strengths, weaknesses and appropriateness of nature-inspired algorithms.	Apply	1	1, 2, 5, 12
3	Apply nature-inspired algorithms to optimization, design and learning problems	Apply	1	1, 2, 3, 5, 12
4	Analyze the Behavior systems of nature inspired algorithm applied in real world problems.	Analyze	2	1, 2, 12
5	Understand the theory behind the design of immune networks and DNA computing and their potential applications.	Analyze	2	1, 2

TEXT BOOK:

1. Leandro Nunes de Castro, " Fundamentals of Natural Computing, Basic Concepts, Algorithms and Applications", Chapman & Hall/ CRC, Taylor and Francis Group, 2007.

REFERENCE BOOKS:

1. Floreano D. and Mattiussi C., "Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies", MIT Press, Cambridge, MA, 2008.
2. Albert Y.Zomaya, "Handbook of Nature-Inspired and Innovative Computing", Springer, 2006.
3. Marco Dorrigo, Thomas Stutzle," Ant Colony Optimization", PHI,2005.

22AM808 SOFT COMPUTING

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Design and Analysis of Algorithms.

COURSE DESCRIPTION AND OBJECTIVES:

Soft computing is an emerging approach to computing which parallel the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. Soft computing is based on some biological inspired methodologies such as genetics, evolution, ant's behaviors, particles swarming, human nervous systems, etc. Now, soft computing is the only solution when we don't have any mathematical modeling of problem solving (i.e., algorithm), need a solution to a complex problem in real time, easy to adapt with changed scenario and can be implemented with parallel computing. It has enormous applications in many application areas such as medical diagnosis, computer vision, hand written character recondition, pattern recognition, machine intelligence, weather forecasting, network optimization, VLSI design, etc.

MODULE-1

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

INTRODUCTION:

Concept of computing systems, "Soft" computing versus "Hard" computing, Characteristics of Soft computing, Some applications of Soft computing

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

FUZZY LOGIC:

Fuzzy Sets, Operations on Fuzzy Sets, Fuzzy Relations, Membership Functions: Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems, Fuzzy Expert Systems, Fuzzy Decision Making.

PRACTICES:

- The membership functions for the linguistic variables "tall" and "short" are given below.

$$\text{tall} = \{\frac{0.2}{5} + \frac{0.3}{7} + \frac{0.7}{9} + \frac{0.9}{11} + \frac{1.0}{12}\}$$
$$\text{short} = \{\frac{0.3}{0} + \frac{0}{30} + \frac{1}{60} + \frac{0.5}{90} + \frac{0}{120}\}$$

  Develop the membership functions for the following linguistic phrases: (a) Very Tall (b) Fairly tall (c) Not very short
- Presence a network of fuzzy inference system.

MODULE-2

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

GENETIC ALGORITHMS:

Goals of optimization, comparison with traditional methods, schemata, Terminology in GA – strings, structure, parameter string, data structures, operators, coding fitness function, algorithm, applications of GA in Machine Learning: Machine Learning Approach to Knowledge Acquisition.

SKILLS:

- ✓ *Skills to generate Frameworks for fuzzy logic and its application*
- ✓ *Construct genetic models for hard problems*
- ✓ *Evaluate and apply*

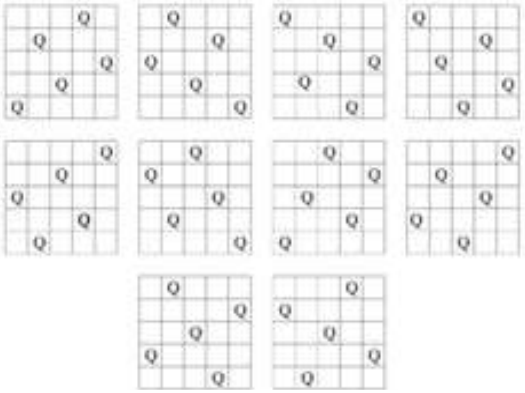
UNIT-2 **12L+8T+0P=20 Hours**

HYBRID SOFT COMPUTING TECHNIQUES AND APPLICATIONS:

Neuro-fuzzy hybrid systems - genetic neuro hybrid systems - genetic fuzzy hybrid and fuzzy genetic hybrid systems - simplified fuzzy ARTMAP - Applications: A fusion approach of multispectral images with SAR, optimization of traveling salesman problem using genetic algorithm approach, soft computing based hybrid fuzzy controllers.

PRACTICES:

- Solve the five queen's problems using Genetic algorithm.



- Solve the TSP problem using Genetic 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 real world problems and identify soft computing techniques suitable for solving them	Analyze	1	1, 2
2	Understand soft computing techniques and their roles in building intelligent machines	Understand	1	1, 2, 3
3	Apply fuzzy logic and reasoning to handle uncertainty and solve various engineering problems	Apply	1	1, 2, 3
4	Use various tools to solve soft computing problems.	Implement	2	11, 12

TEXT BOOKS:

1. D. K. Pratihar, "Soft Computing", Narosa, 2008.
2. Jyh Shing Roger Jang, Chuen Tsai Sun, Eiji Mizutani, Neuro-Fuzzy and Soft Computing, Prentice Hall of India, 2003.

REFERENCE BOOKS:

1. Kasabov, Nikola K, "Foundations of neural networks, fuzzy systems, and knowledge engineering". Marcel Alencar, 1996.
2. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 1995.
3. Timothy J.Ross, Fuzzy Logic with Engineering Applications, 3rd edition, McGraw-Hil, 2011.

22AM809 INTRODUCTION TO SOFTWARE ENGINEERING

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Data Base Management Systems, Oops through Java.

COURSEDESCRIPTIONANDOBJECTIVES:

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 **12L+0T+8P=20Hours**

INTRODUCTION TO SOFTWARE ENGINEERING:

Introduction to Software and Software engineering, evolving role of software, Changing nature of software, 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 assessment.

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

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

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.

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.*

MODULE - 2

UNIT–1 **12L+0T+8P=20 Hours**

TESTING STRATEGIES:

A strategic approach to software testing, Unit testing, Integration testing, Validation testing, System testing, Art of debugging

Testing Tactics: Black-Box testing- Graph based testing, Equivalence partitioning, Boundary value analysis, White-Box testing – basis path testing, Control structure testing.

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

UNIT–2 **12L+0T+8P=20 Hours**

RISK MANAGEMENT:

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

Quality Management: Quality concepts, Software quality assurance, Software reviews, Statistical Software Quality Assurance, ISO 9000 quality standards.

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 shouldsupport 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.

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 Usecase, 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

HONOURS

COMPUTER SCIENCE AND ENGINEERING - DATA SCIENCE

B.Tech.

- 22CS812 - Optimization Techniques
- 22DS805 - Visual Analytics
- 22DS807 - Kernel Methods for Pattern Analysis
- 22CS909 - Capstone Project

COURSE CONTENTS

I SEM & II SEM

22CS812 OPTIMIZATION TECHNIQUES

Hours Per Week :

L	T	P	C
3	0	2	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+0T+8P=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+0T+8P=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.
- 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	

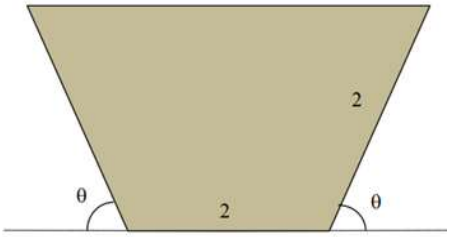
- Find the positive root of the following equation by (Bisection) method,

  $f(x) = \sin\left(\frac{x}{2\pi}\right) - \cos^2 x$

- 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, π/2] find the interval after 3 iterations. Use an initial interval ε = 0.2



MODULE-2

UNIT-1 **12L+0T+8P=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+0T+8P=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 ≤ xi ≤ 1 for i = 1, . . . , n and let P denote the solution set. Explain how to solve a linear programming problem
- max{c T x : x ε P}.
- What if the linear system was ai ≤ xi ≤ bi for i = 1, . . . , n. Here we assume ai ≤ bi 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 → 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.	Under-stand	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	Imple-ment	2	3, 4, 11, 12

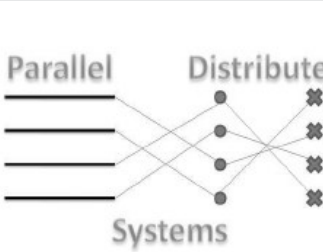
TEXT BOOKS:

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.

22DS805 VISUAL ANALYTICS



Source: http://uceou.edu/PDS/About%20PDS.html

Hours Per Week :

L	T	P	C
3	2	0	4

PREREQUISITE KNOWLEDGE: Introduction to data mining, probability theory and statistics, calculus, matrix algebra and Python Programming.

COURSE DESCRIPTION AND OBJECTIVES:

This course will cover the fundamentals of Visual Analytics. This course provides an understanding of the visual analytics in terms of data. It discusses different principles of Interactive visualisation, Computational techniques in visual analytics and processing of data.

MODULE-1

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

INTRODUCTION TO VISUAL ANALYTICS

What is visual analytics, A motivating example, Discussion and General definitions of visual analytics. **General Concepts:** Subjects of analysis, Structure of an analysis subject, using data to understand a subject, Summary of all general concepts.

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

PRINCIPLES OF INTERACTIVE VISUALISATION

Preliminary notes, Visualisation, Interaction, Summary.

Computational Techniques in Visual Analytics: Visualisation for supporting computations, Computations for supporting visual analysis, Distance functions, Feature selection, Data embedding, clustering, Topic modelling, Conclusion.

PRACTICES:

- Analytical reasoning techniques
- Data processing and model building
- Visual representation and interaction
- Presentation and dissemination
- Group project: Problem can be decided by the team members with prior approval by the course instructor.

MODULE-2

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

VISUAL ANALYTICS FOR INVESTIGATING AND PROCESSING DATA

Examples of data properties that may affect data analysis, investigating data properties: Overall view of a distribution, outliers, missing data, Impacts of data collection and integration procedures, Processing data: Data clustering, modelling for data preparation, Transformation of data elements, Synthesis of data components, data integration, transformation of data structure, and data reduction and selection, conclusion.

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

VISUAL ANALYTICS FOR UNDERSTANDING MULTIPLE ATTRIBUTES

Motivating Example, Specifics of multivariate data, analytical Goals and Tasks, Visual Analytics

Techniques: Analysis characteristics of multiple attributes, analysing multivariate relations, analysing higher-order relations, and local structures

Visual analytics for understanding relationships between entities: Motivating example: Extracting relationships, visualising relationships and exploring relationships, Graphs as a mathematical concept, Graph/network visualization techniques, common tasks in graph/network analysis.

PRACTICES:

- Individual Project: Problem can be decided by the team members with prior approval by the course instructor.

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 the kind of problems visual analytics can solve.	Apply	1	1, 2, 12
2	Implement the techniques practically to have hands on experience for all the methods.	Apply	1	1, 2, 5, 12
3	Develop analytical questions for a data analysis problem and develop appropriate data manipulations and graphs to answer them	Apply	1	1, 2, 3, 5, 12
4	Perform exploratory data analysis and avoid major fallacies of analysis and interpretation, design develop and evaluate effective visual interfaces to analyse tabular, textual, temporal, and spatial data	Analyze	2	1, 2, 3, 12
5	Going in depth to find out research outcomes.	Analyze	2	1, 3, 5, 2

TEXT BOOKS:

1. Natalia Andrienko Gennady Andrienko, Georg Fuchs Aidan Slingsby, Cagatay Turkay Stefan Wrobel, “Visual Analytics for Data Scientists”, Springer, 2020.

REFERENCE BOOKS:

1. Qingyu Zhang, Richard S. Segall, Mei Cao, “Visual Analytics and Interactive Technologies: Data, Text and Web Mining Applications”, IGI Global.
2. Sang C. Suh, Thomas Anthony, “Big Data and Visual Analytics”, Springer, 2017.

22DS807 KERNEL METHODS FOR PATTERN ANALYSIS

Hours Per Week :

L	T	P	C
3	0	2	4

PREREQUISITE KNOWLEDGE: Machine Learning, Python programming.

COURSE DESCRIPTION AND OBJECTIVES:

The course kernel methods for pattern analysis (KMPA) deals with the advanced topics in pattern analysis. Pattern analysis mainly involves pattern classification, regression, and pattern clustering. This course primarily focuses on pattern analysis of data that is often not vectors of numbers. For example, protein sequences and structures in computational biology, text and XML documents in web mining, time series in finance, have structures which contain relevant information for the statistical problem but can hardly be encoded into finite-dimensional vector representations. Kernel methods are a class of algorithms well suited for such problems. Indeed, they extend the applicability of many statistical methods initially designed for vectors to virtually any type of data, without the need for explicit vectorization of the data. The goal of this course is to present the mathematical foundations of kernel methods, as well as the main approaches that have emerged so far in kernel design.

MODULE-1

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

INTRODUCTION AND BACKGROUND

Introduction to Pattern Analysis: Pattern analysis tasks, Pattern analysis techniques, Learning methods

NON-KERNEL METHODS FOR PATTERN ANALYSIS: Polynomial curve fitting, Linear model for regression, Regularization, Bias-variance decomposition, Multilayer feed forward neural network, Auto-associative neural network, Radial basis function neural networks.

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

DEEP GENERATIVE MODELS

Boltzmann Machines, Restricted Boltzmann machines, Deep Belief Networks, Deep Boltzmann Machines, Generative Adversarial Networks.

PRACTICES:

- Logisitic regression model with polynomial basis functions for datasets with (a) Linearly separable classes, (b) Nonlinearly separable classes, (c) Overlapping classes
- MLFFNN with 2 hidden layers using the cross-entropy error function for the datasets with(a) Linearly separable classes, (b) Nonlinearly separable classes, (c) Overlapping classes
- MLFFNN with 3-level stacked RBM based pre-training for Black and white image data
- Visualize Decision region plots for different models specified above
- Plot Confusion matrix for the test data of each dataset and for different models with the best performance.
- Display features extracted by different levels of stacked RBM at the end of pre-training

MODULE-2

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

KERNEL METHODS FOR PATTERN ANALYSIS

Support vector machine, Support vector regression, Support vector data description, v-support vector methods, Kernel K-means clustering, Kernel principal component analysis, Kernel Fisher discriminant analysis, Kernel canonical correlational analysis

Theory of kernels: Reproducing kernel Hilbert space, The Representer theorem, Mercer's theorem, Operations on kernels, Kernels for structured data: Strings, Sets, Graphs and Trees, Kernel learning methods

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

MACHINE LEARNING PARADIGMS FOR PATTERN ANALYSIS

Paradigms in machine learning: Supervised learning, Semi-Supervised learning, Self-Supervised, Unsupervised learning, and Reinforcement learning. Semi-Supervised learning-Self-training method, Graph-based methods, Semi-supervised Gaussian mixture model, Semi-supervised support vector machine .

PRACTICES:

- Implement Models for the following cases:
  - Linear kernel based SVM for a dataset with Linearly separable classes
  - Polynomial kernel-based C-SVM for Datasets with
    (a) Linearly separable classes,
    (b) Nonlinearly separable classes,
    (c) Overlapping classes
  - Gaussian kernel-based C-SVM for Datasets with
    (a) Linearly separable classes,
    (b) Nonlinearly separable classes,
    (c) Overlapping classes
  - Visualize Decision region plots for different models specified above
  - Plot Confusion matrix for the test data of each dataset and for different models with the best performance.
  - Images of kernel gram matrices to justify the choice of kernel parameters for the SVMs with bestperformance, for each kernel and for each dataset as specified above
- Implement Regression Model with v-SVR using Gaussian kernel
- Implement outlier detection Model with v-SVDD using Gaussian kernel
- Implement K-means clustering for a 2-dimensional data of nonlinearly separable classes
- Implement Kernel K-means clustering using Gaussian kernel for a 2-dimensional data of nonlinearly separable classes
- Implement a Classification using kernels for structured data (Graph representation of data)
- Implement a semi-supervised learning approach for classification.

.

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 suitability of kernels for the given supervised and unsupervised problems	Analyze	1,2	2, 4
2	Designing kernels appropriate for given data	Design	1	3
3	Usage of tools to implement complex models	Apply	1,2	1, 5
4	Application of kernels for pattern analysis	Apply	1,2	1

TEXT BOOKS:

1. B. Scholkopf and A.J.Smola, “Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond”, The MIT Press, 2002
2. Ian Good fellow and Yoshua Bengio and Aaron, “Deep Learning”, 1st Edition, An MIT Press Book, 2016

REFERENCE BOOKS:

1. S. Haykin, “Neural Networks and Learning Machines”, 3rd edition, Prentice Hall of India, 2011.
2. Satish Kumar, “Neural Networks, A Classroom Approach”, Tata McGraw -Hill, 2007
3. C.M.Bishop, “Pattern Recognition and Machine Learning”, Cambridge University Press, 2006
4. V.Vapnik, “Statistical Learning Theory”, John Wiley & Sons, 1999.