R23 B.Tech CSE COURSE STRUCTURE & SYLLABUS

From II Year to IV Year

Vision and Mission of the University

Vision

The University is primarily promoting quality of education in the areas of Science, Technology, Engineering and Mathematics (STEM) as four academic pillars of education, to excel in teaching, learning, research, consultancy and placements through innovative practices with global perspective.

Mission

1. Design an Industry relevant curriculum from time to time with a Global perspective
2. Promoting quality education by embracing ICT delivery mechanism with continuous pedagogy through e-learning mechanism
3. Spread across for industry collaborations with a focus to pre-training and placements for technology transfer to society
4. Establishing centers of excellence to promote research and innovations in multidisciplinary areas to bring in patent culture and consultancy practices
5. International Collaborations for student outreach
6. Facilitating international students to study in JNTUK to infuse cross culture learning practices.

Vision and Mission of the Institute

Vision and Mission of the Department

Programme Education Objectives (PEOs) of the B.Tech (CSE)

PEO 1: "Competent IT professional with sound fundamental and applied knowledge in Computer Science and Engineering."

PEO 2: "Sustained learner to bring out creative and innovative ideas to meet the challenges of industry and society with ethics and human values and pursue higher studies."

PEO 3: "Entrepreneurs in computer science acquainted with interpersonal, managerial skills to make them successful in multidisciplinary fields."

Programme Outcomes (POs)

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 modeling to complex engineering activities with an understanding of the limitations.

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

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

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

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

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

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

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

Programme Specific Outcomes (PSOs)

PSO 1: "Design algorithms for real-world computational problems and analyze the complexities to optimize solutions."

PSO 2: "Apply methods and tools for data acquisition, analysis, and knowledge discovery in developing decision-making systems."

PSO 3: "Develop adaptable and self-learning models with real-world data fitting, evaluation, and performance improvement methods."

Mapping of Programme Specific Outcomes to PEOs

PEO \ PSO	PSO1	PSO2	PSO3
PEO1	H	M	M
PEO2	M	H	H
PEO3	L	M	H

Model Lab Examination Rubrics

Criterion	Weightage	Excellent (H) – 8–10	Good (M) – 5–7	Satisfactory (L) – 2–4	Unsatisfactory – 0–1
1. Problem Understanding & Algorithm Design	20%	Demonstrates complete understanding; designs optimal and structured logic/algorithm	Understands problem clearly; logic mostly correct with minor gaps	Partial understanding; logic incomplete or inefficient	No understanding; illogical or irrelevant approach
2. Program Design & Coding Standards	20%	Code is well-structured, modular, uses OOP concepts correctly; excellent naming and formatting	Good structure; correct use of major concepts; minor formatting issues	Basic program structure; limited modularity; poor naming/formatting	Disorganized code; missing structure; no use of OOP concepts
3. Program Execution & Debugging Skills	25%	Program runs perfectly for all cases; independently debugs errors efficiently	Program runs with minor errors; needs minimal help	Program runs partially; needs moderate assistance	Program does not run; unable to debug
4. Output Accuracy & Test Cases	20%	All outputs correct; handled multiple test cases; edge cases included	Outputs mostly correct; some test cases missing	Limited accuracy; few test cases	Incorrect/no outputs
5. Viva Voce (Concepts and topics)	15%	Answers all questions correctly with clarity and depth	Answers most questions correctly with reasonable understanding	Answers basic questions only; limited conceptual clarity	Unable to answer; no conceptual understanding

B.Tech.– II Year I Semester

S.No.	Category	Title	L	T	P	Credits
1	BS&H	Discrete Mathematics & Graph Theory	3	0	0	3
2	BS&H	Universal human values – understanding harmony and Ethical human conduct	2	1	0	3
3	Engineering Science	Digital Logic &Computer Organization	3	0	0	3
4	Professional Core	Advanced Data Structures & Algorithm Analysis	3	0	0	3
5	Professional Core	Object Oriented Programming Through Java	3	0	0	3
6	Professional Core	Advanced Data Structures and Algorithm Analysis Lab	0	0	3	1.5
7	Professional Core	Object Oriented Programming Through Java Lab	0	0	3	1.5
8	Skill Enhancement Course	Python Programming	0	1	2	2
9	Audit Course	Environmental Science	2	0	0	-
Total			**16**	**2**	**8**	**20**

B.Tech.– II Year II Semester

S.No.	Category	Title	L	T	P	Credits
1	Management Course- I	Managerial Economics and Financial Analysis	2	0	0	2
2	Engineering Science/ Basic Science	Probability & Statistics	3	0	0	3
3	Professional Core	Operating Systems	3	0	0	3
4	Professional Core	Database Management Systems	3	0	0	3
5	Professional Core	Software Engineering	2	1	0	3
6	Professional Core	Operating Systems Lab	0	0	3	1.5
7	Professional Core	Database Management Systems Lab	0	0	3	1.5
8	Skill Enhancement Course	Full Stack Development –I	0	1	2	2
9	BS&H	Design Thinking &Innovation	1	0	2	2
Total			**14**	**2**	**10**	**21**
Mandatory Community Service Project Internship of 08 weeks duration during summer vacation						

II Year I Semester

L	T	P	C
3	0	0	3

DISCRETE MATHEMATICS AND GRAPH THEORY

Course Objectives:

- To introduce the students to the topics and techniques of discrete methods and combinatorial reasoning.
- To introduce a wide variety of applications. The algorithmic approach to the solution of problems is fundamental in discrete mathematics, and this approach reinforces the close ties between this discipline and the area of computer science.

Course Outcomes: At the end of the course, successful student will be able to
CO1: Apply the principles of propositional and predicate logic to construct well-formed formulas and validate arguments using inference rules (K3)
CO2: Analyze and solve problems related to sets, relations, functions, lattices, and algebraic structures using appropriate mathematical techniques (K4)
CO3: Apply combinatorial techniques and generating functions to formulate and solve recurrence relations in discrete structures (K3)
CO4: Examine graph theoretic concepts such as paths, circuits, connectivity, and graph isomorphism, and apply them to model and solve real-world problems (K4)
CO5: Apply advanced graph algorithms (BFS, DFS, spanning trees, colouring, planar graphs) and evaluate their suitability in solving computational problems (K5)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	L	L	-	-	-	-	-	-	M	-	-	-
CO2	H	H	L	L	L	-	-	-	-	-	-	M	-	-	-
CO3	H	H	M	M	L	-	-	-	-	-	-	M	-	-	-
CO4	H	H	M	M	M	-	-	-	-	-	-	M	-	-	-
CO5	H	H	H	M	H	-	-	-	-	-	-	M	-	-	-

UNIT–I: Mathematical Logic:

Propositional Calculus: Statements and Notations, Connectives, Well Formed Formulas, Truth Tables, Tautologies, Equivalence of Formulas, Duality Law, Tautological Implications, Normal Forms, Theory of Inference for Statement Calculus, Consistency of Premises, Indirect Method of Proof, Predicate Calculus: Predicates, Predicative Logic, Statement Functions, Variables and Quantifiers, Free and Bound Variables, Inference Theory for Predicate Calculus.

UNIT-II: Set Theory:
Sets: Operations on Sets, Principle of Inclusion-Exclusion, Relations: Properties, Operations, Partition and Covering, Transitive Closure, Equivalence, Compatibility and Partial Ordering, Hasse Diagrams, Functions: Bijective, Composition, Inverse, Permutation, and Recursive Functions, Lattice and its Properties.

UNIT-III: Combinatorics and Recurrence Relations:

Basis of Counting, Permutations, Permutations with Repetitions, Circular and Restricted Permutations, Combinations, Restricted Combinations, Binomial and Multinomial Coefficients and Theorems.

Recurrence Relations:

Generating Functions, Function of Sequences, Partial Fractions, Calculating Coefficient of Generating Functions, Recurrence Relations, Formulation as Recurrence Relations, Solving Recurrence Relations by Substitution and Generating Functions, Method of Characteristic Roots, Solving Inhomogeneous Recurrence Relations

UNIT-IV: Graph Theory:

Basic Concepts, Graph Theory and its Applications, Subgraphs, Graph Representations: Adjacency and Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs,

Unit-V: Multi Graphs

Multigraphs, Bipartite and Planar Graphs, Euler's Theorem, Graph Colouring and Covering, Chromatic Number, Spanning Trees, Prim's and Kruskal's Algorithms, BFS and DFS Spanning Trees.

TEXT BOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science, J. P. Tremblay and P. Manohar, Tata McGraw Hill.
2. Elements of Discrete Mathematics-A Computer Oriented Approach, C. L.Liu and D. P. Mohapatra, 3rd Edition, Tata McGraw Hill.
3. Theory and Problems of Discrete Mathematics, Schaum's Outline Series, Seymour Lipschutz and Marc Lars Lipson, 3rd Edition, McGraw Hill.

REFERENCE BOOKS:

1. Discrete Mathematics for Computer Scientists and Mathematicians, J. L.Mott, A. Kandel and T. P. Baker, 2nd Edition, Prentice Hall of India.
2. Discrete Mathematical Structures, Bernand Kolman, Robert C. Busby andSharon Cutler Ross, PHI.
3. Discrete Mathematics, S. K. Chakraborthy and B.K. Sarkar, Oxford, 2011.
4. Discrete Mathematics and its Applications with Combinatorics and GraphTheory, K. H. Rosen, 7th Edition, Tata McGraw Hill.

II Year I Semester

L	T	P	C
2	1	0	3

UNIVERSAL HUMAN VALUES – UNDERSTANDING HARMONY AND ETHICAL HUMAN CONDUCT

Course Objectives:
- To help the students appreciate the essential complementary between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
- To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. Such holistic perspective forms the basis of Universal Human Values and movement towards value-based living in a natural way.
- To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behaviour and mutually enriching interaction with Nature.

Course Outcomes:
1. Define the terms like Natural Acceptance, Happiness and Prosperity (L1, L2)
2. Identify one's self, and one's surroundings (family, society nature) (L1, L2)
3. Apply what they have learnt to their own self in different day-to-day settings in real life (L3)
4. Relate human values with human relationship and human society. (L4)
5. Justify the need for universal human values and harmonious existence (L5)
6. Develop as socially and ecologically responsible engineers (L3, L6)

Course Topics

The course has 28 lectures and 14 tutorials in 5 modules. The lectures and tutorials are of 1-hour duration. Tutorial sessions are to be used to explore and practice what has been proposed during the lecture sessions.

The Teacher's Manual provides the outline for lectures as well as practice sessions. The teacher is expected to present the issues to be discussed as propositions and encourage the students to have a dialogue.

UNIT I   Introduction to Value Education (6 lectures and 3 tutorials for practice session)

Lecture 1: Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education)
Lecture 2: Understanding Value Education
Tutorial 1: Practice Session PS1 Sharing about Oneself
Lecture 3: self-exploration as the Process for Value Education

Lecture4: Continuous Happiness and Prosperity – the Basic Human Aspirations

Tutorial 2: Practice Session PS2 Exploring Human Consciousness

Lecture 5: Happiness and Prosperity – Current Scenario

Lecture 6: Method to Fulfill the Basic Human Aspirations

Tutorial 3: Practice Session PS3 Exploring Natural Acceptance

UNIT II   Harmony in the Human Being (6 lectures and 3 tutorials for practice session)

Lecture 7: Understanding Human being as the Co-existence of the self and the body.

Lecture 8: Distinguishing between the Needs of the self and the body

Tutorial 4: Practice Session PS4 Exploring the difference of Needs of self and body.

Lecture 9: The body as an Instrument of the self

Lecture 10: Understanding Harmony in the self

Tutorial 5: Practice Session PS5 Exploring Sources of Imagination in the self

Lecture 11: Harmony of the self with the body

Lecture 12: Programme to ensure self-regulation and Health

Tutorial 6: Practice Session PS6 Exploring Harmony of self with the body

UNIT III   Harmony in the Family and Society (6 lectures and 3 tutorials for practice session)

Lecture 13: Harmony in the Family – the Basic Unit of Human Interaction

Lecture 14: 'Trust' – the Foundational Value in Relationship

Tutorial 7: Practice Session PS7 Exploring the Feeling of Trust

Lecture 15: 'Respect' – as the Right Evaluation

Tutorial 8: Practice Session PS8 Exploring the Feeling of Respect

Lecture 16: Other Feelings, Justice in Human-to-Human Relationship

Lecture 17: Understanding Harmony in the Society

Lecture 18: Vision for the Universal Human Order

Tutorial 9: Practice Session PS9 Exploring Systems to fulfil Human Goal

UNIT IV   Harmony in the Nature/Existence (4 lectures and 2 tutorials for practice
session)

Lecture 19: Understanding Harmony in the Nature

Lecture 20: Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature

Tutorial 10: Practice Session PS10 Exploring the Four Orders of Nature

Lecture 21: Realizing Existence as Co-existence at All Levels

Lecture 22: The Holistic Perception of Harmony in Existence

Tutorial 11: Practice Session PS11 Exploring Co-existence in Existence.

UNIT V   Implications of the Holistic Understanding – a Look at Professional Ethics (6 lectures and 3 tutorials for practice session)
Lecture 23: Natural Acceptance of Human Values
Lecture 24: Definitiveness of (Ethical) Human Conduct
Tutorial 12: Practice Session PS12 Exploring Ethical Human Conduct
Lecture 25: A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order
Lecture 26: Competence in Professional Ethics
Tutorial 13: Practice Session PS13 Exploring Humanistic Models in Education
Lecture 27: Holistic Technologies, Production Systems and Management Models-Typical Case Studies
Lecture 28: Strategies for Transition towards Value-based Life and Profession
Tutorial 14: Practice Session PS14 Exploring Steps of Transition towards Universal Human Order

Practice Sessions for UNIT I – Introduction to Value Education
PS1 Sharing about Oneself
PS2 Exploring Human Consciousness
PS3 Exploring Natural Acceptance

Practice Sessions for UNIT II – Harmony in the Human Being
PS4 Exploring the difference of Needs of self and body
PS5 Exploring Sources of Imagination in the self
PS6 Exploring Harmony of self with the body

Practice Sessions for UNIT III – Harmony in the Family and Society
PS7 Exploring the Feeling of Trust
PS8 Exploring the Feeling of Respect
PS9 Exploring Systems to fulfil Human Goal

Practice Sessions for UNIT IV – Harmony in the Nature (Existence)
PS10 Exploring the Four Orders of Nature
PS11 Exploring Co-existence in Existence

Practice Sessions for UNIT V – Implications of the Holistic Understanding – a Look at Professional Ethics
PS12 Exploring Ethical Human Conduct
PS13 Exploring Humanistic Models in Education
PS14 Exploring Steps of Transition towards Universal Human Order

READINGS:
Textbook and Teachers Manual

a. The Textbook

R R Gaur, R Asthana, G P Bagaria, *A Foundation Course in Human Values and Professional Ethics*, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1

b. The Teacher's Manual

R R Gaur, R Asthana, G P Bagaria,*Teachers' Manual for A Foundation Course in Human Values and Professional Ethics,* 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-53-2

Reference Books

1. *JeevanVidya: EkParichaya,* A Nagaraj, JeevanVidyaPrakashan, Amarkantak, 1999.
2. *Human Values*, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. *The Story of Stuff* (Book).
4. *The Story of My Experiments with Truth* - by Mohandas Karamchand Gandhi
5. *Small is Beautiful* - E. F Schumacher.
6. *Slow is Beautifu*l - Cecile Andrews
7. *Economy of Permanence* - J C Kumarappa
8. *Bharat Mein Angreji Raj* – PanditSunderlal
9. *Rediscovering India* - by Dharampal
10. *Hind Swaraj or Indian Home Rule* - by Mohandas K. Gandhi
11. *India Wins Freedom* - Maulana Abdul Kalam Azad
12. *Vivekananda* - Romain Rolland (English)
13. *Gandhi* - Romain Rolland (English)

Mode of Conduct:

Lecture hours are to be used for interactive discussion, placing the proposals about the topics at hand and motivating students to reflect, explore and verify them.

Tutorial hours are to be used for practice sessions.

While analyzing and discussing the topic, the faculty mentor's role is in pointing to essential elements to help in sorting them out from the surface elements. In other words, help the students explore the important or critical elements.

In the discussions, particularly during practice sessions (tutorials), the mentor encourages the student to connect with one's own self and do self-observation, self-reflection and self-exploration.

Scenarios may be used to initiate discussion. The student is encouraged to take up "ordinary" situations rather than" extra-ordinary" situations. Such observations and their analyses are shared and discussed with other students and faculty mentor, in a group sitting.

Tutorials (experiments or practical) are important for the course. The difference is that the laboratory is everyday life, and practical are how you behave and work in real life. Depending on the nature of topics, worksheets, home assignment and/or activity are included. The practice sessions (tutorials) would also provide support to a student in performing actions commensurate to his/her beliefs. It is intended that this would lead to development of commitment, namely behaving and working based on basic human values.

It is recommended that this content be placed before the student as it is, in the form of a basic foundation course, without including anything else or excluding any part of this content. Additional content may be offered in separate, higher courses. This course is to be taught by faculty from every teaching department, not exclusively by any one department.

Teacher preparation with a minimum exposure to at least one 8-day Faculty Development Program on Universal Human Values is deemed essential.

Online Resources:

1. https://fdp-si.aicte-india.org/UHV-II%20Class%20Notes%20&%20Handouts/UHV%20Handout%201-Introduction%20to%20Value%20Education.pdf

2. https://fdp-si.aicte-india.org/UHV-II%20Class%20Notes%20&%20Handouts/UHV%20Handout%202-Harmony%20in%20the%20Human%20Being.pdf

3. https://fdp-si.aicte-india.org/UHV-II%20Class%20Notes%20&%20Handouts/UHV%20Handout%203-Harmony%20in%20the%20Family.pdf

4. https://fdp-si.aicte-india.org/UHV%201%20Teaching%20Material/D3-S2%20Respect%20July%202023.pdf

5. https://fdp-si.aicte-india.org/UHV-II%20Class%20Notes%20&%20Handouts/UHV%20Handout%205-Harmony%20in%20the%20Nature%20and%20Existence.pdf

6. https://fdp-si.aicte-india.org/download/FDPTeachingMaterial/3-days%20FDP-SI%20UHV%20Teaching%20Material/Day%203%20Handouts/UHV%203D%20D3-S2A%20Und%20Nature-Existence.pdf

7. https://fdp-si.aicte-india.org/UHV%20II%20Teaching%20Material/UHV%20II%20Lecture%202023-25%20Ethics%20v1.pdf

8. https://www.studocu.com/in/document/kiet-group-of-institutions/universal-human-values/chapter-5-holistic-understanding-of-harmony-on-professional-ethics/62490385

https://onlinecourses.swayam2.ac.in/aic22_ge23/preview

II Year I Semester

L	T	P	C
3	0	0	3

DIGITAL LOGIC & COMPUTER ORGANIZATION

Course Objectives:

The main objectives of the course is to

- provide students with a comprehensive understanding of digital logic design principles and computer organization fundamentals
- Describe memory hierarchy concepts
- Explain input/output (I/O) systems and their interaction with the CPU, memory, and peripheral devices

Course Outcomes (COs): At the end of the course, successful student will be able to

CO1: *Apply* number systems, binary codes, logic gates, and K-map techniques to *design and simplify* combinational logic circuits. **(K3)**

CO2: *Analyze* the behavior and operation of sequential circuits, flip-flops, counters, and registers, and *explain* the basic structure and performance factors of computer systems. **(K4)**

CO3: *Perform and evaluate* arithmetic operations including addition, subtraction, multiplication, and division using various algorithms, and *examine* processor organization and instruction execution mechanisms. **(K4)**

CO4: *Analyze and compare* different memory organizations, including RAM, ROM, cache, and virtual memory, with respect to speed, size, cost, and performance. **(K4)**

CO5: *Explain and assess* I/O organization concepts such as interrupts, DMA, buses, and interface circuits, and *evaluate* standard I/O interfaces used in modern computer systems. **(K5)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	H	M	L	-	-	-	-	-	-				
CO2	H	H	M	M	L	-	-	-	-	-	-				
CO3	H	M	M	M	L	-	-	-	-	-	-				
CO4	H	H	M	H	M	-	-	-	-	-	-				
CO5	H	H	M	M	M	-	-	-	-	-	-				

UNIT – I:

Data Representation: Binary Numbers, Fixed Point Representation. Floating Point Representation. Number base conversions, Octal and Hexadecimal Numbers, components, Signed binary numbers, Binary codes

Digital Logic Circuits-I: Basic Logic Functions, Logic gates, universal logic gates, Minimization of Logic expressions. K-Map Simplification, Combinational Circuits, Decoders, Multiplexers

UNIT – II:

Digital Logic Circuits-II: Sequential Circuits, Flip-Flops, Binary counters, Registers, Shift Registers, Ripple counters

Basic Structure of Computers: Computer Types, Functional units, Basic operational concepts, Bus structures, Software, Performance, multiprocessors and multi computers, Computer Generations, Von- Neumann Architecture

UNIT – III:
Computer Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Fast Multiplication, Integer Division, Floating-Point Numbers and Operations
Processor Organization: Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control and Multi programmed Control

UNIT – IV:
The Memory Organization: Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage

UNIT – V:
Input/Output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces

Textbooks:
1. Computer Organization, Carl Hamacher, ZvonkoVranesic, SafwatZaky, 6th edition, McGraw Hill
2. Digital Design, 6th Edition, M. Morris Mano, Pearson Education.
3. Computer Organization and Architecture, William Stallings, 11thEdition, Pearson.

Reference Books:
1. Computer Systems Architecture, M.Moris Mano, 3rdEdition, Pearson
2. Computer Organization and Design, David A. Paterson, John L.Hennessy, Elsevier
3. Fundamentals of Logic Design, Roth, 5thEdition, Thomson

Online Learning Resources:
1. https://nptel.ac.in/courses/106/103/106103068/

L	T	P	C
3	0	0	3

II Year I Semester

ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS

Course Objectives:

The main objectives of the course is to

- provide knowledge on advance data structures frequently used in Computer Science domain
- Develop skills in algorithm design techniques popularly used
- Understand the use of various data structures in the algorithm design

Course Outcomes (COs)

CO1: *Analyze* algorithm performance using space and time complexity measures, apply asymptotic notations, and *construct and manipulate* AVL Trees and B-Trees for various applications. **(K4)**

CO2: *Apply and evaluate* heap structures, graph representations, and graph traversal techniques, and *implement* divide-and-conquer algorithms such as quick sort, merge sort, Strassen's multiplication, and convex hull computations. **(K4)**

CO3: *Apply* greedy and dynamic programming strategies to solve optimization problems such as MST, shortest paths, knapsack, OBST, TSP, and string editing, and *analyze* their computational efficiency. **(K4)**

CO4: *Develop and analyze* solutions for combinatorial problems using backtracking and branch-and-bound techniques, including 8-queens, subset sum, graph coloring, knapsack, and TSP. **(K4)**

CO5: *Explain and evaluate* NP-hard and NP-complete problem classes, including graph and scheduling decision problems, and *assess* their computational intractability with reference to theoretical foundations. **(K4)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	L	-	-	-	-	-	-	-			
CO2	H	H	H	M	M	-	-	-	-	-	-	-			
CO3	H	H	H	M	M	-	-	-	-	-	-	-			
CO4	H	M	H	M	L	-	-	-	-	-	-	-			
CO5	H	M	M	M	L	-	-	-	-	-	-	-			

UNIT – I:

Introduction to Algorithm Analysis, Space and Time Complexity analysis, Asymptotic Notations. AVL Trees – Creation, Insertion, Deletion operations and Applications; B-Trees – Creation, Insertion, Deletion operations and Applications

UNIT – II:

Heap Trees (Priority Queues) – Min and Max Heaps, Operations and Applications
Graphs – Terminology, Representations, Basic Search and Traversals, Connected Components and Biconnected Components, applications
Divide and Conquer: The General Method, Quick Sort, Merge Sort, Strassen's matrix multiplication, Convex Hull

UNIT – III:

Greedy Method: General Method, Job Sequencing with deadlines, Knapsack Problem, Minimum cost spanning trees, Single Source Shortest Paths

Dynamic Programming: General Method, All pairs shortest paths, Single Source Shortest Paths– General Weights (Bellman Ford Algorithm), Optimal Binary Search Trees, 0/1 Knapsack, String Editing, Travelling Salesperson problem

UNIT – IV:

Backtracking: General Method, 8-Queens Problem, Sum of Subsets problem, Graph Coloring, 0/1 Knapsack Problem

Branch and Bound: The General Method, 0/1 Knapsack Problem, Travelling Salesperson problem

UNIT – V:

NP Hard and NP Complete Problems: Basic Concepts, Cook's theorem

NP Hard Graph Problems: Clique Decision Problem (CDP), Chromatic Number Decision Problem (CNDP), Traveling Salesperson Decision Problem (TSP)

NP Hard Scheduling Problems: Scheduling Identical Processors, Job Shop Scheduling

Textbooks:
1. Fundamentals of Data Structures in C++, Horowitz, Ellis; Sahni, Sartaj; Mehta, Dinesh, 2nd Edition Universities Press
2. Computer Algorithms in C++, Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, 2nd Edition University Press

Reference Books:
1. Data Structures and program design in C, Robert Kruse, Pearson Education Asia
2. An introduction to Data Structures with applications, Trembley& Sorenson, McGraw Hill
3. The Art of Computer Programming, Vol.1: Fundamental Algorithms, Donald E Knuth, Addison-Wesley, 1997.
4. Data Structures using C & C++: Langsam, Augenstein&Tanenbaum, Pearson, 1995
5. Algorithms + Data Structures & Programs:, N.Wirth, PHI
6. Fundamentals of Data Structures in C++: Horowitz Sahni& Mehta, Galgottia Pub.
7. Data structures in Java:, Thomas Standish, Pearson Education Asia

Online Learning Resources:
1. https://www.tutorialspoint.com/advanced_data_structures/index.asp
2. http://peterindia.net/Algorithms.html
3. Abdul Bari,Introduction to Algorithms (youtube.com)

L	T	P	C
3	0	0	3

II Year I Semester

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Course Objectives:

The learning objectives of this course are to:

- identify Java language components and how they work together in applications
- Learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries.
- learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications
- understand how to design applications with threads in Java
- understand how to use Java APIs for program development

Course Outcomes (COs)

CO1: *Apply* object-oriented programming principles and Java program structure to write, compile, and execute basic Java programs using appropriate data types, operators, and control statements. **(K3)**

CO2: *Develop and utilize* classes, objects, constructors, methods, access specifiers, and method overloading/overriding to implement modular and reusable Java applications. **(K3)**

CO3: *Implement and analyze* arrays, inheritance models, interfaces, and abstract classes to design structured and extensible Java solutions for given problems. **(K4)**

CO4: *Use and evaluate* Java packages, standard libraries, exception handling mechanisms, and Java I/O streams to build robust and efficient applications. **(K4)**

CO5: *Develop and integrate* advanced Java features including string handling, multithreading, JDBC-based database connectivity, and JavaFX GUI components to create interactive and concurrent applications. **(K5)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	M	L	L	-	-	-	-	-	-	-			
CO2	H	M	H	M	L	-	-	-	-	-	-	-			
CO3	H	H	H	M	L	-	-	-	-	-	-	-			
CO4	H	M	M	H	M	-	-	-	-	-	-	-			
CO5	H	M	H	H	M	-	-	-	-	-	-	-			

UNIT I

Object Oriented Programming: Basic concepts, Principles, Program Structure in Java: Introduction, Writing Simple Java Programs, Elements or Tokens in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences Comments, Programming Style.

Data Types, Variables, and Operators :Introduction, Data Types in Java, Declaration of Variables, Data Types, Type Casting, Scope of Variable Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf() Method, Static Variables and Methods, Attribute Final, **Introduction to Operators**, Precedence and Associativity of Operators, Assignment Operator (=), Basic Arithmetic Operators, Increment (++) and Decrement (- -)

Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators.

Control Statements: Introduction, if Expression, Nested if Expressions, if–else Expressions, Ternary Operator?:, Switch Statement, Iteration Statements, while Expression, do–while Loop, for Loop, Nested for Loop, For–Each for Loop, Break Statement, Continue Statement.

UNIT II

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing Private Members of Class, Constructor Methods for Class, Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by Reference, Keyword this.

Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static.

UNIT III

Arrays: Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Sorting of Arrays, Search for Values in Arrays, Class Arrays, Two-dimensional Arrays, Arrays of Varying Lengths, Three-dimensional Arrays, Arrays as Vectors.

Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class-Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Inheritance.

Interfaces: Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface, Functional Interfaces, Annotations.

UNIT IV

Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in Java SE, Java.lang Package and its Classes, Class Object, Enumeration, class Math, Wrapper Classes, Auto-boxing and Auto-unboxing, Java util Classes and Interfaces, Formatter Class, Random Class, Time Package, Class Instant (java.time.Instant), Formatting for Date/Time in Java, Temporal Adjusters Class, Temporal Adjusters Class.

Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions.

Java I/O and File: Java I/O API, standard I/O streams, types, Byte streams, Character streams, Scanner class, Files in Java(Text Book 2)

UNIT V

String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings,Comparison, Modifying, Searching; Class String Buffer.

Multithreaded Programming: Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread-Creation of New Threads, Thread States, Thread Priority-Synchronization, Deadlock and Race Situations, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads.

Java Database Connectivity: Introduction, JDBC Architecture, Installing MySQL and MySQL Connector/J, JDBC Environment Setup, Establishing JDBC Database Connections, Result Set Interface

Java FX GUI: Java FX Scene Builder, Java FX App Window Structure, displaying text and image, event handling, laying out nodes in scene graph, mouse events (Text Book 3)

Text Books:
1) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford.
2) Joy with JAVA, Fundamentals of Object Oriented Programming, DebasisSamanta, MonalisaSarma, Cambridge, 2023.
3) JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

References Books:
1) The complete Reference Java, 11th edition, Herbert Schildt,TMH
2) Introduction to Java programming, 7th Edition, Y Daniel Liang, Pearson

Online Resources:
1) https://nptel.ac.in/courses/106/105/106105191/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618 816347_shared/overview

L	T	P	C
0	0	3	1.5

II Year I Semester

ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS LAB

Course Objectives:

The objectives of the course is to

- acquire practical skills in constructing and managing Data structures
- apply the popular algorithm design methods in problem-solving scenarios

Experiments covering the Topics:

- Operations on AVL trees, B-Trees, Heap Trees
- Graph Traversals
- Sorting techniques
- Minimum cost spanning trees
- Shortest path algorithms
- 0/1 Knapsack Problem
- Travelling Salesperson problem
- Optimal Binary Search Trees
- N-Queens Problem
- Job Sequencing

CO1: *Implement and manipulate* advanced tree data structures such as AVL Trees, B-Trees, and Heaps to perform operations including insertion, deletion, traversal, and file/array-based storage. **(K3)**

CO2: *Develop and analyze* graph traversal and graph-based algorithms such as BFS, DFS, connected and biconnected components using multiple graph representations. **(K4)**

CO3: *Apply and compare* divide-and-conquer, greedy, and dynamic programming techniques to solve problems such as sorting, shortest paths, job sequencing, and knapsack, including performance evaluation for various inputs. **(K4)**

CO4: *Design and implement* backtracking and branch-and-bound solutions for combinatorial optimization problems such as N-Queens, 0/1 Knapsack, and Travelling Salesperson Problem. **(K5)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	M	L	M	-	-	-	L	-	-	L			
CO2	H	H	M	M	M	-	-	-	L	-	-	L			
CO3	H	H	H	M	M	-	-	-	L	-	-	M			
CO4	H	H	H	M	M	-	-	-	L	-	-	M			

Sample Programs:

1. Construct an AVL tree for a given set of elements which are stored in a file. And implement insert and delete operation on the constructed tree. Write contents of tree into a new file using in-order.
2. Construct B-Tree an order of 5 with a set of 100 random elements stored in array. Implement searching, insertion and deletion operations.
3. Construct Min and Max Heap using arrays, delete any element and display the content of the Heap.

4. Implement BFT and DFT for given graph, when graph is represented by
   a) Adjacency Matrix   b) Adjacency Lists
5. Write a program for finding the biconnected components in a given graph.
6. Implement Quick sort and Merge sort and observe the execution time for various input sizes (Average, Worst and Best cases).
7. Compare the performance of Single Source Shortest Paths using Greedy method when the graph is represented by adjacency matrix and adjacency lists.
8. Implement Job Sequencing with deadlines using Greedy strategy.
9. Write a program to solve 0/1 Knapsack problem Using Dynamic Programming.
10. Implement N-Queens Problem Using Backtracking.
11. Use Backtracking strategy to solve 0/1 Knapsack problem.
12. Implement Travelling Sales Person problem using Branch and Bound approach.

Reference Books:

1. Fundamentals of Data Structures in C++, Horowitz Ellis, SahniSartaj, Mehta, Dinesh, 2nd Edition, Universities Press
2. Computer Algorithms/C++ Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, 2nd Edition, University Press
3. Data Structures and program design in C, Robert Kruse, Pearson Education Asia
4. An introduction to Data Structures with applications, Trembley& Sorenson, McGraw Hill

Online Learning Resources:

1. http://cse01-iiith.vlabs.ac.in/
2. http://peterindia.net/Algorithms.html

L	T	P	C
0	0	3	1.5

II Year I Semester

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

Course Objectives:

The aim of this course is to

- Practice object oriented programming in the Java programming language
- Implement Classes, Objects, Methods, Inheritance, Exception, Runtime Polymorphism, User defined Exception handling mechanism
- Illustrate inheritance, Exception handling mechanism, JDBC connectivity
- Construct Threads, Event Handling, implement packages, Java FX GUI

Course Outcomes:

CO1: Apply object-oriented programming fundamentals—including data types, control structures, classes, methods, inheritance, and polymorphism—to develop structured Java programs (K3)

CO2: Implement and demonstrate Java features such as interfaces, packages, exception handling, and multithreading to build modular, robust, and concurrent applications (K4)

CO3: Use and manage files, I/O streams, JDBC connectivity, and JavaFX GUI components to create interactive applications involving data input, storage, and display (K4)

CO4: Design and evaluate Java applications by integrating OOP concepts, multithreading, exceptions, GUI and database operations to solve real-world programming problems (K5)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	H	L	L	-	-	-	-	-	-	-			
CO2	H	M	H	H	L	-	-	-	-	-	-	-			
CO3	H	M	M	H	M	-	-	-	-	-	-	-			
CO4	H	H	H	H	M	-	-	-	-	-	-	-			

Experiments covering the Topics:

- Object Oriented Programming fundamentals- data types, control structures
- Classes, methods, objects, Inheritance, polymorphism,
- Exception handling, Threads, Packages, Interfaces
- Files, I/O streams, Java FX GUI

Sample Experiments:

Exercise – 1:

a) Write a JAVA program to display default value of all primitive data type of JAVA

b) Write a java program that display the roots of a quadratic equation $ax^2+bx=0$. Calculate the discriminate D and basing on value of D, describe the nature of root.

Exercise - 2

a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism.

b) Write a JAVA program to sort for an element in a given list of elements using bubble sort

c) Write a JAVA program using String Buffer to delete, remove character.

Exercise - 3

a) Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.

b) Write a JAVA program implement method overloading.

c) Write a JAVA program to implement constructor.

d)Write a JAVA program to implement constructor overloading.

Exercise - 4

a) Write a JAVA program to implement Single Inheritance

b) Write a JAVA program to implement multi level Inheritance

c) Write a JAVA program for abstract class to find areas of different shapes

Exercise - 5

a) Write a JAVA program give example for "super" keyword.

b) Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?

c) Write a JAVA program that implements Runtime polymorphism

Exercise - 6

a) Write a JAVA program that describes exception handling mechanism

b) Write a JAVA program Illustrating Multiple catch clauses

c) Write a JAVA program for creation of Java Built-in Exceptions

d) Write a JAVA program for creation of User Defined Exception

Exercise - 7

a) Write a JAVA program that creates threads by extending Thread class.First thread display "Good Morning "every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds,(Repeat the same by implementing Runnable)

b) Write a program illustrating **is Alive** and **join ()**

c) Write a Program illustrating Daemon Threads.

d) Write a JAVA program Producer Consumer Problem

Exercise – 8

a) Write a JAVA program that import and use the user defined packages

b) Without writing any code, build a GUI that display text in label and image in an Image View (use JavaFX)

c) Build a Tip Calculator app using several JavaFX components and learn how to respond to user interactions with the GUI

Exercise – 9

a) Write a java program that connects to a database using JDBC

b) Write a java program to connect to a database using JDBC and insert values into it.

c) Write a java program to connect to a database using JDBC and delete values from it

L	T	P	C
0	1	2	2

II Year I Semester

PYTHON PROGRAMMING
(SKILL ENHANCEMENT COURSE)

Course Objectives:

The main objectives of the course are to
- Introduce core programming concepts of Python programming language.
- Demonstrate about Python data structures like Lists, Tuples, Sets and dictionaries
- Implement Functions, Modules and Regular Expressions in Python Programming and to create practical and contemporary applications using these

Course Outcomes:

CO1: Apply core Python programming concepts—including data types, operators, control flow statements, functions, and basic data structures—to solve computational problems efficiently (K3)

CO2: Implement and analyze string operations, list manipulations, tuple and dictionary operations, and set-based concepts to handle structured data in Python programs (K4)

CO3: Use and manage file handling techniques, object-oriented programming principles, and modular programming constructs to develop clean, reusable, and real-world Python applications (K4)

CO4: Develop data handling and data-processing programs using JSON, XML, NumPy, and Pandas, including array operations, slicing, indexing, and tabular data exploration (K5)

CO5: Design and build Python applications integrating OOP, file processing, data structures, functional programming, and basic data science libraries to address real-world computational tasks. (K5)

UNTI-I:

History of Python Programming Language, Thrust Areas of Python, Installing Anaconda Python Distribution, Installing and Using Jupiter Notebook.

Parts of Python Programming Language: Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, the type () Function and Is Operator, Dynamic and Strongly Typed Language.

Control Flow Statements: if statement, if-else statement, if...elif…else, Nested if statement, while Loop, for Loop, continue and break Statements, Catching Exceptions Using try and except Statement.

Sample Experiments:

1. Write a program to find the largest element among three Numbers.
2. Write a Program to display all prime numbers within an interval
3. Write a program to swap two numbers without using a temporary variable.
4. Demonstrate the following Operators in Python with suitable examples.

 i) Arithmetic Operators ii) Relational Operators iii) Assignment Operatorsiv) Logical Operators v) Bit wise Operators vi) Ternary Operator vii) Membership Operatorsviii) Identity Operators

5. Write a program to add and multiply complex numbers
6. Write a program to print multiplication table of a given number.

UNIT-II:

Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the function, return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments.

Strings: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, del Statement.

Sample Experiments:

1. Write a program to define a function with multiple return values.
2. Write a program to define a function using default arguments.
3. Write a program to find the length of the string without using any library functions.
4. Write a program to check if the substring is present in a given string or not.
5. Write a program to perform the given operations on a list:
   > i. Addition, ii. Insertion, iii. slicing
6. Write a program to perform any 5 built-in functions by taking any list.

UNIT-III:

Dictionaries: Creating Dictionary, Accessing and Modifying key:value Pairs in Dictionaries, Built-In Functions Used on Dictionaries, Dictionary Methods, del Statement.

Tuples and Sets: Creating Tuples, Basic Tuple Operations, tuple() Function, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Using zip() Function, Sets, Set Methods, Frozenset.

Sample Experiments:

1. Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenated tuples.
2. Write a program to count the number of vowels in a string (No control flow allowed).
3. Write a program to check if a given key exists in a dictionary or not.
4. Write a program to add a new key-value pair to an existing dictionary.
5. Write a program to sum all the items in a given dictionary.

UNIT-IV:

Files: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules.

Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, Constructor Method, Classes with Multiple Objects, Class Attributes Vs Data Attributes, Encapsulation, Inheritance, Polymorphism.

Sample Experiments:

1. Write a program to sort words in a file and put them in another file. The output file should have only lower-case words, so any upper-case words from source must be lowered.

2. Python program to print each line of a file in reverse order.
3. Python program to compute the number of characters, words and lines in a file.
4. Write a program to create, display, append, insert and reverse the order of the items in the array.
5. Write a program to add, transpose and multiply two matrices.
6. Write a Python program to create a class that represents a shape. Include methods to calculate its area and perimeter. Implement subclasses for different shapes like circle, triangle, and square.

UNIT-V:
Introduction to Data Science: Functional Programming, JSON and XML in Python, NumPy with Python, Pandas.

Sample Experiments:
1. Python program to check whether a JSON string contains complex object or not.
2. Python Program to demonstrate NumPy arrays creation using array () function.
3. Python program to demonstrate use of ndim, shape, size, dtype.
4. Python program to demonstrate basic slicing, integer and Boolean indexing.
5. Python program to find min, max, sum, cumulative sum of array
6. Create a dictionary with at least five keys and each key represent value as a list where this list contains at least ten values and convert this dictionary as a pandas data frame and explore the data through the data frame as follows:
   a) Apply head () function to the pandas data frame
   b) Perform various data selection operations on Data Frame
7. Select any two columns from the above data frame, and observe the change in one attribute with respect to other attribute with scatter and plot operations in matplotlib

Reference Books:
1. Gowrishankar S, Veena A., Introduction to Python Programming, CRC Press.
2. Python Programming, S Sridhar, J Indumathi, V M Hariharan, 2nd Edition, Pearson, 2024
3. Introduction to Programming Using Python, Y. Daniel Liang, Pearson.

Online Learning Resources/Virtual Labs:
1. https://www.coursera.org/learn/python-for-applied-data-science-ai
2. https://www.coursera.org/learn/python?specialization=python#syllabus

II Year I Semester

L	T	P	C
2	0	0	0

ENVIRONMENTAL SCIENCE

Course Objectives:
- To make the students to get awareness on environment
- To understand the importance of protecting natural resources, ecosystems for future generations and pollution causes due to the day to day activities of human life
- To save earth from the inventions by the engineers.

Course Outcomes:
1. Demonstrate an understanding of the multidisciplinary nature of environmental studies and identify various renewable and non-renewable natural resources.
2. Explain the concepts of energy flow in ecosystems, bio-geo-chemical cycles, and ecological pyramids.
3. Analyze the major causes of environmental pollution, principles of solid waste management, and related preventive and control measures.
4. Describe the concepts and significance of rainwater harvesting, watershed management, ozone layer depletion, and wasteland reclamation.
5. Discuss the causes and impacts of population explosion, and understand the importance of value education and various welfare programmes.

UNIT–I

Multidisciplinary Nature Of Environmental Studies: – Definition, Scope and Importance – Need for Public Awareness.

Natural Resources: Renewable and non-renewable resources – Natural resources and associated problems – Forest resources – Use and over – exploitation, deforestation, case studies – Timber extraction – Mining, dams and other effects on forest and tribal people – Water resources – Use and over utilization of surface and ground water – Floods, drought, conflicts over water, dams – benefits and problems–Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies– Food resources: World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies.–Energy resources:

UNIT–II

Ecosystems: Concep to fan ecosystem.–Structure and function of an ecosystem–Producers, consumers and decomposers – Energy flow in the ecosystem – Ecological succession – Food chains, food webs and ecological pyramids–Introduction, types, characteristic features, structure and function of the following ecosystem:
- a. Forest ecosystem.
- b. Grassl and ecosystem
- c. Desert ecosystem
- d. Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)

Biodiversity And Its Conservation: Introduction Definition: genetic, species and ecosystem diversity–Bio-geographical classification of India–Value of biodiversity: consumptive use, Productive use, social, ethical, aesthetic and option values – Biodiversity at global, National and local levels – India as a mega-diversity nation – Hot-sports of biodiversity – Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts– Endangered and endemic species of India –Conservation of biodiversity :In-situand Ex-situ conservation of biodiversity.

UNIT–III

Environmental Pollution: Definition, Cause, effects and control measures of:

a. Air Pollution.
b. Water pollution
c. Soil pollution
d. Marine pollution
e. Noise pollution
f. Thermal pollution
g. Nuclear hazards

Solid Waste Management: Causes, effects and control measures of urban and industrial wastes – Role of an individual in prevention of pollution – Pollution case studies – Disaster management: floods, earthquake, cyclone and landslides.

UNIT–IV

Social Issues and the Environment: From Unsustainable to Sustainable development– Urban problems related to energy – Water conservation, rain water harvesting, watershed management –Resettlement and rehabilitation of people; its problems and concerns. Case studies – Environmental ethics: Issues and possible solutions–Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case Studies – Wastel and reclamation. – Consumerism and waste products. – Environment Protection Act. – Air (Prevention and Control of Pollution) Act. –Water (Prevention and control of Pollution) Act–Wild life Protection Act–Forest Conservation Act–Issues involved inenforcement of environment allegislation–Public awareness.

UNIT–V

Human Population And The Environment: Population growth, variation among nations. Population explosion – Family Welfare Programmes. – Environment and human health – Human Rights – Value Education–HIV/AIDS–Women and Child Welfare–Role of information Technology in Environment and human health–Case studies. Field Work: Visit to a local area to document environmental assets River/ forest grassland/ hill/ mountain – Visit to a local polluted site-Urban/Rural/Industrial/Agricultural Study of common plants, insects, and birds–river, hills lopes, etc..

Text books:

1. Text book of Environmental Studies for Undergraduate Courses ErachBharucha for University Grants Commission, Universities Press.
2. Palaniswamy,"Environmental Studies", Pearson education
3. S.AzeemUnnisa,"Environmental Studies" Academic Publishing Company
4. K.Raghavan Nambiar,"Text book of Environmental Studies for Undergraduate Courses as per UGC model syllabus", Scitech Publications (India), Pvt.Ltd.

Reference Books:

1. Deeksha Dave and E.Sai Baba Reddy, "Text book of Environmental Science", Cengage Publications.
2. M.Anji Reddy, "Text book of Environmental Sciences and Technology" , BS Publication.
3. J.P. Sharma, Comprehensive Environmental studies, Laxmi publications.
4. J.Glynn Henry and GaryW.Heinke,"Environmental Sciences and Engineering", Prentice Hall of India Private limited
5. G.R.Chatwal, "A Text Book of Environmental Studies "Himalaya Publishing House
6. Gilbert M.Masters and Wendell P.Ela, "Introduction to Environmental Engineering and Science, Prentice Hall of India Private limited.

II Year II Semester

L	T	P	C
2	0	0	2

MANAGERIAL ECONOMICS AND FINANCIAL ANALYSIS

Course Objectives:
- To inculcate the basic knowledge of micro economics and financial accounting
- To make the students learn how demand is estimated for different products, input-out put relationship for optimizing production and cost
- To Know the Various types of market structure and pricing methods and strategy
- To give an overview on investment appraisal methods to promote the students to learn how to plan long-term investment decisions.
- To provide fundamental skills on accounting and to explain the process of preparing financial statements.

Course Outcomes:
1. Define the concepts related to Managerial Economics, financial accounting and management.
2. Explain the fundamentals of Economics viz., Demand, Production, cost, revenue and markets
3. Apply the Concept of Production cost and revenues for effective Business decision
4. Analyze how to invest their capital and maximize returns
5. Evaluate the capital budgeting techniques
6. Develop the accounting statements and evaluate the financial performance of business entity.

UNIT-I
Managerial Economics: Introduction – Nature, meaning, significance, functions, and advantages. Demand-Concept, Function,Law of Demand - Demand Elasticity- Types – Measurement. Demand Forecasting- Factors governing Forecasting, Methods. Managerial Economics and Financial Accounting and Management.

UNIT-II
Production and Cost Analysis: Introduction – Nature, meaning, significance, functions and advantages. Production Function– Least-cost combination– Short run and long run Production Function- Isoquants and Isocosts, MRTS -Cobb-Douglas Production Function - Laws of Returns - Internal and External Economies of scale. Cost & Break-Even Analysis - Cost concepts and Cost behaviour- Break-Even Analysis (BEA) -Determination of Break-Even Point (Simple Problems)-Managerial significance and limitations of Break-Even Analysis.

UNIT-III

Business Organizations and Markets: Introduction–Nature, meaning, significance, functions and advantages. Forms of Business Organizations- Sole Proprietary - Partnership - Joint Stock Companies - Public Sector Enterprises. Types of Markets - Perfect and Imperfect Competition - Features of Perfect Competition Monopoly-Monopolistic Competition–Oligopoly-Price-Output Determination-Pricing Methods and Strategies

UNIT-IV

Capital Budgeting: Introduction – Nature, meaning, significance, functions and advantages. Types of Working Capital, Components, Sources of Short-term and Long-term Capital, Estimating Working capital requirements. Capital Budgeting–Features, Proposals, Methods and Evaluation. Projects– Pay Back Method, Accounting Rate of Return(ARR) Net Present Value(NPV)Internal Rate Return(IRR)
Method (sample problems)

UNIT-V

Financial Accounting and Analysis: Introduction – Nature, meaning, significance, functions and advantages. Concepts and Conventions-Double-Entry Book Keeping, Journal, Ledger, Trial Balance-Final Accounts (Trading Account, Profit and Loss Account and Balance Sheet with simple adjustments). Financial Analysis-Analysis and Interpretation of Liquidity Ratios, Activity Ratios, and Capital structure Ratios and Profitability.

Textbooks:
1. Varshney & Maheswari: Managerial Economics, Sultan Chand,2013.

Reference Books:
1. Managerial Economics: Principles And Worldwide Applications, 9E (Adaptation) by Dominick Salvatore and Siddhartha Rastogi
2. Managerial Economics: Principles and Worldwide Applications
   by Dominick Salvatore

II Year II Semester

L	T	P	C
3	0	0	3

PROBABILITY AND STATISTICS

Course Objectives:
- To familiarize the students with the foundations of probability and statistical methods
- To impart probability concepts and statistical methods in various applications Engineering

Course Outcomes: Upon successful completion of this course, the student should be able to
1. Classify the concepts of data science and its importance (L2)
2. Interpret the association of characteristics and through correlation and regression tools (L4)
3. Apply discrete and continuous probability distributions (L3)
4. Design the components of a classical hypothesis test   (L6)
5. Infer the statistical inferential methods based on small and large sampling tests (L4)

Unit – I: Descriptive statistics and methods for data science:
Data science – Statistics Introduction – Population vs Sample –Collection of data – primary and secondary data – Type of variable: dependent and independent Categorical and Continuous variables – Data visualization – Measures of Central tendency – Measures of Variability – Skewness – Kurtosis.

UNIT – II: Correlation and Regression:

Correlation – Correlation coefficient – Rank correlation.
Linear Regression: Straight line – Multiple Linear Regression - Regression coefficients and properties **–** Curvilinear Regression: Parabola – Exponential – Power curves.

UNIT – III: Probability and Distributions:

Probability– Conditional probability and Baye's theorem – Random variables – Discrete and Continuous random variables – Distribution functions – Probability mass function, Probability density function and Cumulative distribution functions – Mathematical Expectation and Variance – Binomial, Poisson, Uniform and Normal distributions.

UNIT – IV: Sampling Theory:

Introduction – Population and Samples – Sampling distribution of Means and Variance (definition only) – Point and Interval estimations – Maximum error of estimate – Central limit theorem (without proof) – Estimation using t, χ^2 and F-distributions.

UNIT – V: Tests of Hypothesis:

Introduction – Hypothesis – Null and Alternative Hypothesis – Type I and Type II errors – Level of significance – One tail and two-tail tests – Test of significance for large samples and Small Samples: Single and difference means – Single and two proportions – Student's t- test, F-test, χ^2-test.

Text Books:

- **Miller and Freund's**, Probability and Statistics for Engineers,7/e, Pearson, 2008.

- **S. C. Gupta and V.K. Kapoor**, Fundamentals of Mathematical Statistics, 11/e, Sultan Chand & Sons Publications, 2012.

Reference Books:
- **Shron L. Myers, Keying Ye, Ronald E Walpole,** Probability and Statistics Engineers and the Scientists,8th Edition, Pearson 2007.
- **Jay l. Devore,** Probability and Statistics for Engineering and the Sciences, 8th Edition, Cengage.
- **Sheldon M. Ross**, Introduction to probability and statistics Engineers and the Scientists, 4th Edition, Academic Foundation, 2011.
- **Johannes Ledolter and Robert V. Hogg**, Applied statistics for Engineers and Physical Scientists, 3rd Edition, Pearson, 2010.

II Year II Semester

L	T	P	C
3	0	0	3

OPERATING SYSTEMS

Course Objectives:

The main objectives of the course is to make student

- Understand the basic concepts and principles of operating systems, including process management, memory management, file systems, and Protection
- Make use of process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
- Illustrate different conditions for deadlock and their possible solutions.

Course Outcomes:

CO1: Explain the fundamental concepts, structures, and operational principles of operating systems, including system calls, services, and system design (K2)

CO2: Demonstrate understanding of processes, threads, concurrency, and CPU scheduling algorithms to effectively manage execution in operating systems (K3)

CO3: Apply synchronization mechanisms and analyze deadlock conditions to design safe and concurrent process interactions (K4)

CO4: Analyze and compare memory-management techniques, virtual memory strategies, and storage management approaches used in modern operating systems (K4)

CO5: Explain the concepts of file systems, directory structures, allocation methods, and protection mechanisms to understand secure and efficient data management (K2)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	M	L	L	-	-	-	-	-	-	-			
CO2	H	H	M	M	L	-	-	-	-	-	-	-			
CO3	H	H	H	M	M	-	-	-	-	-	-	-			
CO4	H	M	H	H	M	-	-	-	-	-	-	-			
CO5	H	M	M	M	L	-	-	-	-	-	-	-			

UNIT - I

Operating Systems Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Free and Open-Source Operating Systems

System Structures: Operating System Services, User and Operating-System Interface, system calls, Types of System Calls, system programs, Operating system Design and Implementation, Operating system structure, Building and Booting an Operating System, Operating system debugging

UNIT - II

Processes: Process Concept, Process scheduling, Operations on processes, Inter-process communication.

Threads and Concurrency: Multithreading models, Thread libraries, Threading issues.

CPU Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling.

UNIT – III

Synchronization Tools: The Critical Section Problem, Peterson's Solution, Mutex Locks, Semaphores, Monitors, Classic problems of Synchronization.
Deadlocks: system Model, Deadlock characterization, Methods for handling Deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from Deadlock.

UNIT - IV

Memory-Management Strategies: Introduction, Contiguous memory allocation, Paging, Structure of the Page Table, Swapping.
Virtual Memory Management: Introduction, Demand paging, Copy-on-write, Page replacement, Allocation of frames, Thrashing
Storage Management: Overview of Mass Storage Structure, HDD Scheduling.

UNIT - V

File System: File System Interface: File concept, Access methods, Directory Structure; File system Implementation: File-system structure, File-system Operations, Directory

implementation, Allocation method, Free space management; File-System Internals: File-System Mounting, Partitions and Mounting, File Sharing.
Protection: Goals of protection, Principles of protection, Protection Rings, Domain of protection, Access matrix.

Text Books:
1. Operating System Concepts, Silberschatz A, Galvin P B, Gagne G, 10th Edition, Wiley, 2018.
2. Modern Operating Systems, Tanenbaum A S, 4th Edition, Pearson , 2016

Reference Books:
1. Operating Systems -Internals and Design Principles, Stallings W, 9th edition, Pearson, 2018
2. Operating Systems: A Concept Based Approach, D.M Dhamdhere, 3rd Edition, McGraw- Hill, 2013

Online Learning Resources:
1. https://nptel.ac.in/courses/106/106/106106144/
2. http://peterindia.net/OperatingSystems.html

L	T	P	C
3	0	0	3

II Year II Semester

DATABASE MANAGEMENT SYSTEMS

Course Objectives:

The main objectives of the course is to

- Introduce database management systems and to give a good formal foundation on the relational model of data and usage of Relational Algebra
- Introduce the concepts of basic SQL as a universal Database language
- Demonstrate the principles behind systematic database design approaches by covering conceptual design, logical design through normalization
- Provide an overview of physical design of a database system, by discussing Database indexing techniques and storage techniques

Course Outcomes (COs)

CO1: *Explain* the fundamental concepts of database systems, including characteristics, architecture, data models, and differences between database and file systems. **(K2)**

CO2: *Construct and interpret* Entity-Relationship (ER) diagram, including entities, attributes, relationships, generalization, specialization, and inheritance for real-world scenarios. **(K3)**

CO3: *Apply* the relational model concepts, including tuples, relations, domains, constraints, relational algebra, and relational calculus to model and query databases. **(K3)**

CO4: *Develop and execute* SQL queries for data definition, data manipulation, aggregation, joins, nested queries, views, and relational set operations on a database schema. **(K3)**

CO5: *Analyze and design* normalized database schemas using functional dependencies, various normal forms (1NF–5NF), and schema refinement techniques to ensure minimal redundancy and data integrity. **(K4)**

CO6: *Explain and implement* transaction management, concurrency control, recovery mechanisms, and indexing techniques (B+ Trees, Hash-based indexing) to ensure ACID properties and efficient data access. **(K4)**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	L	-	-	-	-	-	-	-	-			
CO2	H	H	M	M	-	-	-	-	-	-	-	-			
CO3	H	H	H	M	-	-	-	-	-	-	-	-			
CO4	H	H	H	H	-	-	-	-	-	-	-	-			
CO5	H	H	H	H	M	-	-	-	-	-	-	-			
CO6	H	M	H	H	M	-	-	-	-	-	-	-			

UNIT I:

Introduction: Database system, Characteristics (Database Vs File System), Database Users, Advantages of Database systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams.

UNIT II:

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance, Relational Algebra, Relational Calculus. BASIC SQL: Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update).

UNIT III:

SQL: Basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions(Date and Time, Numeric, String conversion).Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view(updatable and non-updatable), relational set operations.

UNIT IV:

Schema Refinement (Normalization): Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency Lossless join and dependency preserving decomposition, (1NF, 2NF and 3 NF), concept of surrogate key, Boyce-Codd normal form(BCNF), MVD, Fourth normal form(4NF), Fifth Normal Form (5NF).

UNIT V:

Transaction Concept: Transaction State, ACID properties, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, lock based, time stamp based, optimistic, concurrency protocols, Deadlocks, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm.
Introduction to Indexing Techniques: B+ Trees, operations on B+Trees, Hash Based Indexing:

Text Books:
1) Database Management Systems, 3rd edition, Raghurama Krishnan, Johannes Gehrke, TMH (For Chapters 2, 3, 4)
2) Database System Concepts,5th edition, Silberschatz, Korth, Sudarsan,TMH (For Chapter 1 and Chapter 5)

Reference Books:

1) Introduction to Database Systems, 8thedition, C J Date, Pearson.
2) Database Management System, 6th edition, RamezElmasri, Shamkant B. Navathe, Pearson
3) Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

Web-Resources:

1) https://nptel.ac.in/courses/106/105/106105175/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012758066672820 22456_shared/overview

L	T	P	C
2	1	0	3

II Year II Semester

SOFTWARE ENGINEERING

Course Objectives:

The objectives of this course are to introduce

- Software life cycle models, Software requirements and SRS document.
- Project Planning, quality control and ensuring good quality software.
- Software Testing strategies, use of CASE tools, Implementation issues, validation &verification procedures.

Course Outcomes (COs)

CO1: Explain the evolution of software engineering, software development projects, and various software life cycle models, including Waterfall, Spiral, Agile, and RAD. (K2)

CO2: Analyze and apply software project management principles, estimation techniques, risk management, and software requirements gathering to plan and manage software projects effectively. (K4)

CO3: Design software systems using structured analysis, function-oriented design, layered modules, cohesion and coupling concepts, and develop effective user interfaces. (K4)

CO4: Implement, test, and evaluate software through coding standards, black-box and white-box testing, debugging, integration testing, and quality management practices, ensuring software reliability. (K4)

CO5: Apply and evaluate computer-aided software engineering (CASE) tools, software maintenance strategies, and software reuse approaches to enhance productivity and maintainability in software development. (K4)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	L	-	-	-	-	-	-	-	-			
CO2	H	H	M	M	-	-	-	-	-	-	-	-			
CO3	H	H	H	H	M	-	-	-	-	-	-	-			
CO4	H	M	H	H	M	-	-	-	-	-	-	-			
CO5	H	M	H	H	M	-	-	-	-	-	-	-			

UNIT I:

Introduction: Evolution, Software development projects, Exploratory style of software developments, Emergence of software engineering, Notable changes in software development practices, Computer system engineering.

Software Life Cycle Models: Basic concepts, Waterfall model and its extensions, Rapid application development, Agile development model, Spiral model.

UNIT II:

Software Project Management: Software project management complexities, Responsibilities of a software project manager, Metrics for project size estimation, Project

estimation techniques, Empirical Estimation techniques, COCOMO, Halstead's software science, risk management.

Requirements Analysis And Specification: Requirements gathering and analysis, Software Requirements Specification (SRS), Formal system specification, Axiomatic specification, Algebraic specification, Executable specification and 4GL.

UNIT III:
Software Design: Overview of the design process, How to characterize a good software design? Layered arrangement of modules, Cohesion and Coupling. approaches to software design.

Agility: Agility and the Cost of Change, Agile Process, Extreme Programming (XP), Other Agile Process Models, Tool Set for the Agile Process (Text Book 2)

Function-Oriented Software Design: Overview of SA/SD methodology, Structured analysis, Developing the DFD model of a system, Structured design, Detailed design, and Design Review.

User Interface Design: Characteristics of a good user interface, Basic concepts, Types of user interfaces, Fundamentals of component-based GUI development, and user interface design methodology.

UNIT IV:
Coding And Testing: Coding, Code review, Software documentation, Testing, Black-box testing, White-Box testing, Debugging, Program analysis tools, Integration testing, Testing object-oriented programs, Smoke testing, and Some general issues associated with testing.

Software Reliability And Quality Management: Software reliability. Statistical testing, Software quality, Software quality management system, ISO 9000.SEI Capability maturity model. Few other important quality standards, and Six Sigma.

UNIT V:
Computer-Aided Software Engineering (Case): CASE and its scope, CASE environment, CASE support in the software life cycle, other characteristics of CASE tools, Towards second generation CASE Tool, and Architecture of a CASE Environment.

Software Maintenance: Characteristics of software maintenance, Software reverse engineering, Software maintenance process models and Estimation of maintenance cost.

Software Reuse: reuse- definition, introduction, reason behind no reuse so far, Basic issues in any reuse program, A reuse approach, and Reuse at organization level.

Text Books:
1. Fundamentals of Software Engineering, Rajib Mall, 5th Edition,PHI.
2. Software Engineering A practitioner's Approach, Roger S. Pressman, 9th Edition, Mc-Graw Hill International Edition.

Reference Books:
1. Software Engineering, Ian Sommerville,10th Edition, Pearson.
2. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.

e-Resources:
1) https://nptel.ac.in/courses/106/105/106105182/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012605895063871 48827_shared/overview
3) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013382690411003 904735_shared/overview

L	T	P	C
0	0	3	1.5

II Year II Semester

OPERATING SYSTEMS LAB

Course Objectives:
The main objectives of the course are to

- Provide insights into system calls, file systems, semaphores,
- Develop and debug CPU Scheduling algorithms, page replacement algorithms, thread implementation
- Implement Bankers Algorithms to Avoid the Dead Lock

Experiments covering the Topics:

- UNIX fundamentals, commands & system calls
- CPU Scheduling algorithms, thread processing
- IPC, semaphores, monitors, deadlocks
- Page replacement algorithms, file allocation strategies
- Memory allocation strategies

Sample Experiments:

1. Practicing of Basic UNIX Commands.
2. Write programs using the following UNIX operating system calls
   fork, exec, getpid, exit, wait, close, stat, opendir and readdir
3. Simulate UNIX commands like cp, ls, grep, etc.,
4. Simulate the following CPU scheduling algorithms
   a) FCFS b) SJF c) Priority d) Round Robin
5. Control the number of ports opened by the operating system with
   a) Semaphore b) Monitors.
6. Write a program to illustrate concurrent execution of threads using pthreads library.
7. Write a program to solve producer-consumer problem using Semaphores.
8. Implement the following memory allocation methods for fixed partition
   a) First fit b) Worst fit c) Best fit
9. Simulate the following page replacement algorithms
   a) FIFO b) LRU c) LFU
10. Simulate Paging Technique of memory management.
11. Implement Bankers Algorithm for Dead Lock avoidance and prevention
12. Simulate the following file allocation strategies
    a) Sequential b) Indexed c) Linked
13. Download and install nachos operating system and experiment with it

Reference Books:

1. Operating System Concepts, Silberschatz A, Galvin P B, Gagne G, 10th Edition, Wiley, 2018.
2. Modern Operating Systems, Tanenbaum A S, 4th Edition, Pearson, 2016
3. Operating Systems -Internals and Design Principles, Stallings W, 9th edition, Pearson, 2018
4. Operating Systems: A Concept Based Approach, D.M Dhamdhere, 3rd Edition, McGraw- Hill, 2013

Online Learning Resources:

1. https://www.cse.iitb.ac.in/~mythili/os/
2. http://peterindia.net/OperatingSystems.html
3. www.cs.washington.edu/~tom/nachos

L	T	P	C
0	0	3	1.5

II Year II Semester

DATABASE MANAGEMENT SYSTEMS LAB

Course Objectives:

This Course will enable students to

- Populate and query a database using SQL DDL/DML Commands
- Declare and enforce integrity constraints on a database
- Writing Queries using advanced concepts of SQL
- Programming PL/SQL including procedures, functions, cursors and triggers

Experiments covering the topics:

- DDL, DML, DCL commands
- Queries, nested queries, built-in functions,
- PL/SQL programming- control structures
- Procedures, Functions, Cursors, Triggers,
- Database connectivity- ODBC/JDBC

Sample Experiments:

1. Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
2. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Example:- Select the roll number and name of the student who secured fourth rank in the class.
3. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
4. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)
5.
    i. Create a simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
    ii. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.
6. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
7. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE-APPLICATION ERROR.

8. Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
9. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
10. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
11. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers
12. Create a table and perform the search operation on table using indexing and non-indexing techniques.
13. Write a Java program that connects to a database using JDBC
14. Write a Java program to connect to a database using JDBC and insert values into it
15. Write a Java program to connect to a database using JDBC and delete values from it

Text Books/Suggested Reading:
1. Oracle: The Complete Reference by Oracle Press
2. Nilesh Shah, "Database Systems Using Oracle", PHI, 2007
3. Rick F Vander Lans, "Introduction to SQL", Fourth Edition, Pearson Education, 2007

L	T	P	C
0	1	2	2

II Year II Semester

FULL STACK DEVELOPMENT – 1
(SKILL ENHANCEMENT COURSE)

Course Objectives:
The main objectives of the course are to
- Make use of HTML elements and their attributes for designing static web pages
- Build a web page by applying appropriate CSS styles to HTML elements
- Experiment with JavaScript to develop dynamic web pages and validate forms

Experiments covering the Topics:
- Lists, Links and Images
- HTML Tables, Forms and Frames
- HTML 5 and Cascading Style Sheets, Types of CSS
- Selector forms
- CSS with Color, Background, Font, Text and CSS Box Model
- Applying JavaScript - internal and external, I/O, Type Conversion
- JavaScript Conditional Statements and Loops, Pre-defined and User-defined Objects
- JavaScript Functions and Events
- Node.js

Sample Experiments:

1. Lists, Links and Images
a. Write a HTML program, to explain the working of lists.
   Note: It should have an ordered list, unordered list, nested lists and ordered list in an unordered list and definition lists.
b. Write a HTML program, to explain the working of hyperlinks using <a> tag and href, target Attributes.
c. Create a HTML document that has your image and your friend's image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles.
d. Write a HTML program, in such a way that, rather than placing large images on a page, the preferred technique is to use thumbnails by setting the height and width parameters to something like to 100*100 pixels. Each thumbnail image is also a link to a full sized version of the image. Create an image gallery using this technique

2. HTML Tables, Forms and Frames
a. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)

b. Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.).

c. Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes, radio buttons, list boxes using <select>&<option> tags, <text area> and two buttons ie: submit and reset. Use tables to provide a better view).

d. Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: first frame → image, second frame → paragraph, third frame → hyperlink. And also make sure of using "no frame" attribute such that frames to be fixed).

3. HTML 5 and Cascading Style Sheets, Types of CSS

a. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, <div>, <span> tags.

b. Write a HTML program, to embed audio and video into HTML web page.

c. Write a program to apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).

4. Selector forms

a. Write a program to apply different types of selector forms
  i.   Simple selector (element, id, class, group, universal)
  ii. Combinator selector (descendant, child, adjacent sibling, general sibling)
  iii. Pseudo-class selector
  iv. Pseudo-element selector
  v.   Attribute selector

5. CSS with Color, Background, Font, Text and CSS Box Model

a. Write a program to demonstrate the various ways you can reference a color in CSS.

b. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.

c. Write a program using the following terms related to CSS font and text:
  i. font-size   ii. font-weight   iii. font-style
  iv. text-decoration   v. text-transformation   vi. text-alignment

d. Write a program, to explain the importance of CSS Box model using
  i. Content   ii. Border   iii. Margin   iv. padding

6. Applying JavaScript - internal and external, I/O, Type Conversion

a. Write a program to embed internal and external JavaScript in a web page.

b. Write a program to explain the different ways for displaying output.

c. Write a program to explain the different ways for taking input.

d. Create a webpage which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not

7. Java Script Pre-defined and User-defined Objects

a. Write a program using document object properties and methods.
b. Write a program using window object properties and methods.
c. Write a program using array object properties and methods.
d. Write a program using math object properties and methods.
e. Write a program using string object properties and methods.
f. Write a program using regex object properties and methods.
g. Write a program using date object properties and methods.
h. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.

8. Java Script Conditional Statements and Loops

a. Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words "LARGER NUMBER" in an information message dialog. If the numbers are equal, output HTML text as "EQUAL NUMBERS".
b. Write a program to display week days using switch case.
c. Write a program to print 1 to 10 numbers using for, while and do-while loops.
d. Write aprogram to print data in object using for-in, for-each and for-of loops
e. Develop a program to determine whether a given number is an 'ARMSTRONG NUMBER' or not. [Eg: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e.,13 + 53+ 33 = 153]
f. Write a program to display the denomination of the amount deposited in the bank in terms of 100's, 50's, 20's, 10's, 5's, 2's & 1's. (Eg: If deposited amount is Rs.163, the output should be 1-100's, 1-50's, 1- 10's, 1-2's & 1-1's)

9. Java Script Functions and Events

a. Design a appropriate function should be called to display
    i. Factorial of that number
    ii. Fibonacci series up to that number
    iii. Prime numbers up to that number
    iv. Is it palindrome or not
b. Design a HTML having a text box and four buttons named Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate function should be called to display
    i. Factorial of that number
    ii. Fibonacci series up to that number
    iii. Prime numbers up to that number
    iv. Is it palindrome or not
c. Write a program to validate the following fields in a registration page

i. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters)
ii. Mobile (only numbers and length 10 digits)
iii. E-mail (should contain format like xxxxxxx@xxxxxx.xxx)

Text Books:
1. Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson, 2013.
2. Web Programming with HTML5, CSS and JavaScript, John Dean, Jones & Bartlett Learning, 2019 (Chapters 1-11).
3. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.

Web Links:

1. https://www.w3schools.com/html
2. https://www.w3schools.com/css
3. https://www.w3schools.com/js/
4. https://www.w3schools.com/nodejs

II Year II Semester

L	T	P	C
1	0	2	2

DESIGN THINKING & INNOVATION

Course Objectives: The objectives of the course are to

- Bring awareness on innovative design and new product development.
- Explain the basics of design thinking.
- Familiarize the role of reverse engineering in product development.
- Train how to identify the needs of society and convert into demand.
- Introduce product planning and product development process.

UNIT – I   Introduction to Design Thinking

Introduction to elements and principles of Design, basics of design-dot, line, shape, form as fundamental design components. Principles of design. Introduction to design thinking, history of Design Thinking, New materials in Industry.

UNIT - II   Design Thinking Process

Design thinking process (empathize, analyze, idea & prototype), implementing the process in driving inventions, design thinking in social innovations. Tools of design thinking - person, costumer, journey map, brainstorming, product development

Activity: Every student presents their idea in three minutes, Every student can present design process in the form of flow diagram or flow chart etc. Every student should explain about product development.

UNIT - III   Innovation

Art of innovation, Difference between innovation and creativity, role of creativity and innovation in organizations. Creativity to Innovation. Teams for innovation, Measuring the impact and value of creativity.

Activity: Debate on innovation and creativity, Flow and planning from idea to innovation, Debate on value-based innovation.

UNIT - IV   Product Design

Problem formation, introduction to product design, Product strategies, Product value, Product planning, product specifications. Innovation towards product design Case studies.

Activity: Importance of modeling, how to set specifications, Explaining their own product design.

UNIT – V   Design Thinking in Business Processes

Design Thinking applied in Business & Strategic Innovation, Design Thinking principles that redefine business – Business challenges: Growth, Predictability, Change, Maintaining Relevance, Extreme competition, Standardization. Design thinking to meet corporate needs. Design thinking for Startups. Defining and testing Business Models and Business Cases. Developing & testing prototypes.

Activity: How to market our own product, about maintenance, Reliability and plan for startup.

Textbooks:

1. Tim Brown, Change by design, 1/e, Harper Bollins, 2009.
2. Idris Mootee, Design Thinking for Strategic Innovation, 1/e, Adams Media, 2014.

Reference Books:

1. David Lee, Design Thinking in the Classroom, Ulysses press, 2018.
2. Shrrutin N Shetty, Design the Future, 1/e, Norton Press, 2018.
3. William lidwell, Kritinaholden, &Jill butter, Universal principles of design, 2/e, Rockport Publishers, 2010.
4. Chesbrough.H, The era of open innovation, 2003.

Online Learning Resources:

- https://nptel.ac.in/courses/110/106/110106124/
- https://nptel.ac.in/courses/109/104/109104109/
- https://swayam.gov.in/nd1_noc19_mg60/preview
- https://onlinecourses.nptel.ac.in/noc22_de16/preview

Course Outcomes:

COs	Statements	Level
CO1	Define the concepts related to design thinking.	L1
CO2	Explain the fundamentals of Design Thinking and innovation.	L2
CO3	Apply the design thinking techniques for solving problems in various sectors.	L3
CO4	Analyse to work in a multidisciplinary environment.	L4
CO5	Evaluate the value of creativity.	L5

B.Tech. – III Year I Semester

S.No	Category	Title	L	T	P	C
1	Professional Core	Data Warehousing & Data Mining	3	0	0	3
2	Professional Core	Computer Networks	3	0	0	3
3	Professional Core	Formal Languages and Automata Theory	3	0	0	3
4	Professional Elective-I	1. Object Oriented Analysis and Design 2. Artificial Intelligence 3. Microprocessors & Microcontrollers 4. Quantum Computing 5. 12 week MOOC Swayam/ NPTEL course recommended by the BoS	3	0	0	3
5	Open Elective-I	OR Entrepreneurship Development & Venture Creation	3	0	0	3
6	Professional Core	Data Mining Lab	0	0	3	1.5
7	Professional Core	Computer Networks Lab	0	0	3	1.5
8	Skill Enhancement course	Full Stack Development-2	0	1	2	2
9	Engineering Science	User Interface Design using Flutter / SWAYAM Plus - Android Application Development (with Flutter)	0	0	2	1
10	Evaluation of Community Service Internship		-	-	-	2
	Total		15	1	10	23
MC	Minor Course (Student may select from the same specialized minors pool)		3	0	3	4.5
MC	Minor Course through SWAYAM/NPTEL (minimum 12 week, 3 credit course)		3	0	0	3
HC	Honors Course (Student may select from the same honors pool)		3	0	0	3
HC	Honors Course (Student may select from the same honors pool)		3	0	0	3

B.Tech. III Year II Semester

S.No	Category	Title	L	T	P	C
1	Professional Core	Compiler Design	3	0	0	3
2	Professional Core	Cloud Computing	3	0	0	3
3	Professional Core	Cryptography & Network Security	3	0	0	3
4	Professional Elective-II	1. Software Testing Methodologies 2. Cyber Security 3. DevOps 4. Machine Learning 5. 12 week MOOC Swayam/NPTEL course recommended by the BoS	3	0	0	3
5	Professional Elective-III	1. Software Project Management 2. Mobile Adhoc Networks 3. Natural Language Processing 4. Big Data Analytics 5. Distributed Operating System 6. 12 week MOOC Swayam/NPTEL course recommended by the BoS	3	0	0	3
6	Open Elective – II		3	0	0	3
7	Professional Core	Cloud Computing Lab	0	0	3	1.5
8	Professional Core	Cryptography & Network Security Lab	0	0	3	1.5
9	Skill Enhancement course	Soft skills / / SWAYAM Plus - 21st Century Employability Skills	0	1	2	2
10	Audit Course	Technical Paper Writing &IPR	2	0	0	-
	Total		**20**	**1**	**08**	**23**
Mandatory Industry Internship / Mini Project of 08 weeks duration during summer vacation						
MC	Minor Course (Student may select from the same specialized minors pool)		3	0	3	4.5
MC	Minor Course (Student may select from the same specialized minors pool)		3	0	0	3
HC	Honors Course (Student may select from the same honors pool)		3	0	0	3
HC	Honors Course (Student may select from the honors pool)		3	0	0	3

* Under Industry Internship interested students can pursue SWAYAM Plus courses viz., Hands-on Masterclass on Data Analytics OR Artificial Intelligence for Real-World Application

Open Electives, offered to other department students:

Open Elective I: Principles of Operating Systems/ Computer Organization and Architecture

Open Elective II: Principles of Database Management Systems

Open Elective III: Object Oriented Programming Through Java

Open Elective IV: Principles of Software Engineering /Computer Networks

Minor Engineering

Note:

1. *To obtain Minor Engineering, student needs to obtain 18 credits by successfully completing any of the following courses in the concern stream.*

2. *During Minor/Honors Course selection, there should not be any overlapping with Regular/Major/OPEN Electives*

Minor in CSE

1. Principles of Database Management Systems   3-0-3-4.5 (II-II)
2. Principles of Software Engineering   3-0-0-3 (III-I)
3. Advanced Data Structures & Algorithm Analysis   3-0-3-4.5 (III-II)
4. Principles of Operating Systems   3-0-0-3 (IV-I)

    Any of the following 12 Week 3 credit NPTEL MOOC Courses

5. Artificial Intelligence: Knowledge Representation and Reasoning
6. Computer Networks and Internet Protocol
7. Machine Learning and Deep Learning - Fundamentals and Applications
8. Fundamentals of Object Oriented Programming
9. Discrete Mathematics for CS
10. Software Engineering

COURSES OFFERED FOR HONORS DEGREE IN CSE

Note: *To obtain Honor's degree, student needs to obtain 18 credits by successfully completing any of the following courses in the concern stream.*

1. Social Network Analysis   12 Week 3 Credit Course, MOOCS
2. Applied Linear Algebra in AI & ML   12 Week 3 Credit Course, MOOCS
3. Design & Implementation of Human-Computer Interfaces – NPTEL MOOCS
4. Cryptography and Network Security   12 Week 3 Credit Course, MOOCS
5. Privacy and Security in Online Social Media 12 Week 3 Credit Course, MOOCS
6. Deep Learning for Natural Language Processing - 12 Week 3 Credit Course, MOOCS
7. Computer Vision   - 12 Week 3 Credit Course, MOOCS
8. Applied Time-Series Analysis   12 Week 3 Credit Course, MOOCS
9. Parallel Computer Architecture   12 Week 3 Credit Course, MOOCS
10. Reinforcement Learning   12 Week 3 Credit Course, MOOCS
11. GPU Architecture and Programming   12 Week 3 Credit Course, MOOCS
12. Computational Complexity   12 Week 3 Credit Course, MOOCS
13. Quantum Algorithms and Cryptography   12 Week 3 Credit Course, MOOCS
14. Unmanned Arial Systems & Robotics   12 Week 3 Credit Course, MOOCS
15. Prompt Engineering for Generative AI   (III - II)

III Year I Semester	DATA WAREHOUSING & DATA MINING	L	T	P	C
		3	0	0	3

Pre-requisites: Data Structures, Algorithms, Probability & Statistics, Data Base Management Systems

Course Objectives: The main objective of the course is to
- Introduce basic concepts and techniques of data warehousing and data mining
- Examine the types of the data to be mined and apply pre-processing methods on raw data
- Discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms.

Course Outcomes (COs)
CO1: Explain the concepts of data warehousing, OLAP, data cubes, cloud data warehouses, and data visualization to support decision-making. (K2)
CO2: Apply data preprocessing techniques—including cleaning, integration, transformation, reduction, and discretization—to prepare datasets for analysis. (K3)
CO3: Analyze classification problems using decision trees, Bayesian and rule-based classifiers, and evaluate models for accuracy and scalability. (K4)
CO4: Develop association rules from transactional data using algorithms such as Apriori and FP-Growth, and perform pruning and compact representation of frequent itemsets (K5)
CO5: Analyze clustering techniques such as K-means, hierarchical clustering, and DBSCAN to group data and interpret the resulting clusters for knowledge discovery. (K4)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	L	-	-	-	-	-	-	-	-			
CO2	H	H	M	M	-	-	-	-	-	-	-	-			
CO3	H	H	H	H	M	-	-	-	-	-	-	-			
CO4	H	H	H	H	M	-	-	-	-	-	-	-			
CO5	H	H	M	H	M	-	-	-	-	-	-	-			

UNIT–I: Data Warehousing and Online Analytical Processing: Basic concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Cloud Data Warehouse, Data Mining and Patten Mining, Technologies, Applications, Major issues, Data Objects & Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity. (Text Book- 1)

UNIT II: Data Preprocessing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization. (Text Book- 1)

UNIT–III: Classification: Basic Concepts, General Approach to solving a classification problem, Decision Tree Induction: Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Visual Mining for Decision Tree Induction, Bayesian

Classification Methods: Bayes Theorem, Naïve Bayes Classification, Rule-Based Classification, Model Evaluation and Selection. (Text Book- 2)

UNIT–IV: Association Analysis: Problem Definition, Frequent Itemset Generation, Rule Generation: Confident Based Pruning, Rule Generation in Apriori Algorithm, Compact Representation of frequent item sets, FP-Growth Algorithm. (Text Book- 2)

UNIT–V: Cluster Analysis: Overview, Basics and Importance of Cluster Analysis, Clustering techniques, Different Types of Clusters; K-means: The Basic K-means Algorithm, K-means Additional Issues, Bi-secting K Means, Agglomerative Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering Algorithm DBSCAN: Traditional Density Center-Based Approach, DBSCAN Algorithm, Strengths and Weaknesses. (Text Book- 2)

Text Books:
1. Data Mining concepts and Techniques, 3rd edition, Jiawei Han, Michel Kamber, Elsevier, 2011.
2. Introduction to Data Mining: Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson, 2012.

Reference Books:
1. Data Mining: VikramPudi and P. Radha Krishna, Oxford Publisher.
2. Data Mining Techniques, Arun K Pujari, 3rd edition, Universities Press,2013.
3. (NPTEL course by Prof.PabitraMitra)
   http://onlinecourses.nptel.ac.in/noc17_mg24/preview
4. http://www.saedsayad.com/data_mining_map.htm

III Year I Semester	COMPUTER NETWORKS	L	T	P	C
		3	0	0	3

Course Objectives:
- To provide insight about networks, topologies, and the key concepts.
- To gain comprehensive knowledge about the layered communication architectures (OSI and TCP/IP) and its functionalities.
- To understand the principles, key protocols, design issues, and significance of each layers in ISO and TCP/IP.
- To know the basic concepts of network services and various network applications.

Course Outcomes: *After successful completion of the course, student will be able to*

CO1: Explain computer network types, network topologies, OSI & TCP/IP reference models, and characteristics of guided and unguided transmission media.

CO2: Analyze data link layer design issues, framing methods, error detection/correction mechanisms, and evaluate sliding-window–based flow control protocols.

CO3: Compare random, controlled, and channelization MAC techniques and examine the operation and evolution of Ethernet technologies.

CO4: Apply routing algorithms and congestion-control techniques to analyze the performance of packet-switched networks.

CO5: Examine IPv4 and IPv6 protocols, addressing schemes, subnetting/CIDR, and internetworking mechanisms such as tunneling and fragmentation.

CO6: Describe the services of transport and application layer protocols (UDP, TCP, HTTP, DNS, Email) and analyze their operational features including flow, error, and congestion control.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	-	-	-	-	-	-	-	M	-	L			
CO2	H	H	-	M	L	-	-	-	-	M	-	L			
CO3	H	M	M	-	L	-	-	-	-	M	-	L			
CO4	H	H	-	H	L	-	-	-	-	M	-	L			
CO5	H	H	L	M	L	-	-	-	-	M	-	L			
CO6	H	M	-	M	L	-	-	-	-	H	-	L			

UNIT I: Introduction: Network Types, LAN, MAN, WAN, Network Topologies Reference models- The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models, OSI Vs TCP/IP.

Physical Layer –Introduction to Guided Media- Twisted-pair cable, Coaxial cable and Fiber optic cable and introduction about unguided media.

UNIT II: Data link layer: Design issues, **Framing**: fixed size framing, variable size framing, flow control, error control, error detection and correction codes, CRC, Checksum: idea, one's complement internet checksum, services provided to Network Layer, **Elementary**

Data Link Layer protocols: simplex protocol, Simplex stop and wait, Simplex protocol for Noisy Channel.
Sliding window protocol: One bit, Go back N, Selective repeat-Stop and wait protocol, Data link layer in HDLC, Point to point protocol (PPP)

UNIT – III: Media Access Control: Random Access: ALOHA, Carrier sense multiple access (CSMA), CSMA with Collision Detection, CSMA with Collision Avoidance, **Controlled Access:** Reservation, Polling, Token Passing, **Channelization:** frequency division multiple Access(FDMA), time division multiple access(TDMA), code division multiple access(CDMA).
Wired LANs: Ethernet, Ethernet Protocol, Standard Ethernet, Fast Ethernet(100 Mbps), Gigabit Ethernet, 10 Gigabit Ethernet.

UNIT – IV: The Network Layer Design Issues – Store and Forward Packet Switching-Services Provided to the Transport layer- Implementation of Connectionless Service-Implementation of Connection Oriented Service- Comparison of Virtual Circuit and Datagram Networks,
Routing Algorithms-The Optimality principle-Shortest path, Flooding, Distance vector, Link state, Hierarchical, Congestion Control algorithms-General principles of congestion control, Congestion prevention polices, Approaches to Congestion Control-Traffic Aware Routing-Admission Control-Traffic Throttling-Load Shedding. Traffic Control Algorithm-Leaky bucket & Token bucket.
Internet Working: How networks differ- How networks can be connected- Tunnelling, internetwork routing-, Fragmentation, network layer in the internet – IP protocols-IP Version 4 protocol-IPV4 Header Format, IP addresses, Class full Addressing, CIDR, Subnets-IP Version 6-The main IPV6 header, Transition from IPV4 to IPV6, Comparison of IPV4 & IPV6.

UNIT –V: The Transport Layer: Transport layer protocols: Introduction-services- port number-User data gram protocol-User datagram-UDP services-UDP applications-Transmission control protocol: TCP services- TCP features- Segment- A TCP connection-windows in TCP- flow control-Error control, Congestion control in TCP.
Application Layer –- World Wide Web: HTTP, Electronic mail-Architecture- web based mail- email security- TELENET-local versus remote Logging-Domain Name System.

Text Books:
1. Computer Networksm, Andrew S Tanenbaum, Fifth Edition. Pearson Education/PHI
2. Data Communications and Networks, Behrouz A. Forouzan, Fifth Edition TMH.

References Books:
1. Data Communications and Networks- Achut S Godbole, AtulKahate
2. Computer Networks, Mayank Dave, CENGAGE

III Year I Semester	FORMAL LANGUAGES AND AUTOMATA THEORY	L	T	P	C
		3	0	0	3

Course Objectives:

- To learn fundamentals of Regular and Context Free Grammars and Languages
- To understand the relation between Regular Language and Finite Automata and machines
- To learn how to design Automata's and machines as Acceptors, Verifiers and Translators
- To understand the relation between Contexts free Languages, PDA and TM
- To learn how to design PDA as acceptor and TM as Calculators

Course Outcomes: *After successful completion of the course, student will be able to*

CO1: Explain the foundational concepts of automata theory and construct various finite automata (DFA, NFA, ε–NFA, Mealy & Moore machines), including minimization and applications.

CO2: Apply regular expressions, regular grammars, pumping lemma, and closure properties to describe and analyze regular languages.

CO3: Develop and simplify context-free grammars, analyze derivations and parse trees, and convert grammars into normal forms while evaluating ambiguity.

CO4: Design pushdown automata and demonstrate their equivalence with context-free grammars for recognizing context-free languages.

CO5: Construct Turing Machines and analyze decidability, undecidability, PCP, the halting problem, and complexity classes (P, NP, NP-hard, NP-complete).

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	M	-	-	-	-	-	-	L	-	L			
CO2	H	H	-	M	L	-	-	-	-	L	-	L			
CO3	H	H	M	M	L	-	-	-	-	L	-	L			
CO4	H	H	M	M	L	-	-	-	-	L	-	L			
CO5	H	H	L	H	L	-	-	-	-	L	-	L			

UNIT I

Finite Automata: Need of Automata theory, Central Concepts of Automata Theory, Automation, Finite Automation, Transition Systems, Acceptance of a String, DFA, Design of DFAs, NFA, Design of NFA, Equivalence of DFA and NFA, Conversion of NFA into DFA, Finite Automata with Є-Transitions, Minimization of Finite Automata, Finite Automata with output-Mealy and Moore Machines, Applications and Limitation of Finite Automata.

UNIT II

Regular Expressions, Regular Sets, Identity Rules, Equivalence of two RE, Manipulations of REs, Finite Automata and Regular Expressions, Inter Conversion, Equivalence between FA and RE, Pumping Lemma of Regular Sets, Closure Properties of Regular Sets, Grammars,

Classification of Grammars, Chomsky Hierarchy Theorem, Right and Left Linear Regular Grammars, Equivalence between RG and FA, Inter Conversion.

UNIT III

Formal Languages, Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars-Elimination of Useless Symbols, Є-Productions and Unit Productions, Normal Forms-Chomsky Normal Form and Greibach Normal Form, Pumping Lemma, Closure Properties, Applications of Context Free Grammars.

UNIT IV

Pushdown Automata, Definition, Model, Graphical Notation, Instantaneous Description, Language Acceptance of Pushdown Automata, Design of Pushdown Automata, Deterministic and Non – Deterministic Pushdown Automata, Equivalence of Pushdown Automata and Context Free Grammars, Conversion, Two Stack Pushdown Automata, Application of Pushdown Automata.

UNIT V

Turning Machine: Definition, Model, Representation of TMs-Instantaneous Descriptions, Transition Tables and Transition Diagrams, Language of a TM, Design of TMs, Types of TMs, Church's Thesis, Universal and Restricted TM, Decidable and Un-decidable Problems, Halting Problem of TMs, Post's Correspondence Problem, Modified PCP, Classes of P and NP, NP-Hard and NP-Complete Problems.

Text Books:

1. Introduction to Automata Theory, Languages and Computation, J. E. Hopcroft, R. Motwani and J. D. Ullman, 3rd Edition, Pearson, 2008

2. Theory of Computer Science-Automata, Languages and Computation, K. L. P. Mishra and N. Chandrasekharan, 3rd Edition, PHI, 2007

Reference Books:

1. Elements of Theory of Computation, Lewis H.P. & Papadimition C.H., Pearson /PHI
2. Theory of Computation, V. Kulkarni, Oxford University Press, 2013
3. Theory of Automata, Languages and Computation, Rajendra kumar, McGraw Hill, 2014

e-Resources:

1) https://nptel.ac.in/courses/106/104/106104028/

III Year I Semester	OBJECT ORIENTED ANALYSIS AND DESIGN	L	T	P	C
		3	0	0	3

Course Objectives: The main objective is the students to

- Become familiar with all phases of OOAD.
- Master the main features of the UML.
- Master the main concepts of Object Technologies and how to apply them at work and develop the ability to analyze and solve challenging problem in various domains.
- Learn the Object design Principles and understand how to apply them towards Implementation.

Course Outcomes: *After successful completion of the course, student will be able to*

CO1: Explain the structure, attributes, and complexities of large software systems and apply principles for designing complex systems (K2)

CO2: Explain UML fundamentals, its conceptual model, modeling principles, and apply basic structural modeling using classes, relationships, and diagrams (K2)

CO3: Develop class and object diagrams by applying advanced structural modeling concepts such as interfaces, packages, roles, and relationships for diverse case studies (K3)

CO4: Construct behavioral models, including use case, interaction, and activity diagrams, to accurately represent system behavior for real-world applications (K3)

CO5: Design advanced behavioral and architectural models using state machines, events, components, and deployment diagrams for complex software systems (K4)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	-	-	-	-	-	-	-	-	L	M	L	-
CO2	H	M	M	-	L	-	-	-	-	-	-	L	M	L	-
CO3	H	M	H	L	M	-	-	-	-	L	-	L	H	L	L
CO4	M	M	H	M	M	-	-	-	-	L	-	M	M	M	L
CO5	M	H	H	M	H	-	-	-	-	L	-	M	H	M	M

UNIT I:

Introduction: The Structure of Complex systems, The Inherent Complexity of Software, Attributes of Complex System, Organized and Disorganized Complexity, Bringing Order to Chaos, Designing Complex Systems. **Case Study:** System Architecture: Satellite-Based Navigation

UNIT II:

Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, and Software Development Life Cycle. **Basic Structural Modeling:** Classes, Relationships, common Mechanisms, and diagrams. **Case Study:** Control System: Traffic Management.

UNIT III:

Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams. **Advanced Structural Modeling:** Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages. **Case Study:** AI: Cryptanalysis.

UNIT IV

Basic Behavioral Modeling-I: Interactions, Interaction diagrams Use cases, Use case Diagrams, Activity Diagrams. **Case Study:** Web Application: Vacation Tracking System

UNIT V:

Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams. **Architectural Modeling:** Component, Deployment, Component diagrams and Deployment diagrams. **Case Study:** Weather Forecasting

Text Books:

1. Grady BOOCH, Robert A. Maksimchuk, Michael W. ENGLE, Bobbi J. Young, Jim Conallen, Kellia Houston , "Object- Oriented Analysis and Design with Applications", 3rd edition, 2013, PEARSON.
2. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education.

Reference Books:

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY- Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

III Year I Semester	ARTIFICIAL INTELLIGENCE	L	T	P	C
		3	0	0	3

Pre-requisite:
1. Knowledge in Computer Programming.
2. A course on "Mathematical Foundations of Computer Science".
3. Background in linear algebra, data structures and algorithms, and probability.

Course Objectives:
1. The student should be made to study the concepts of Artificial Intelligence.
2. The student should be made to learn the methods of solving problems using Artificial Intelligence.
3. The student should be made to introduce the concepts of Expert Systems.
4. To understand the applications of AI, namely game playing, theorem proving, and machine learning.
5. To learn different knowledge representation techniques

Course Outcomes: *After successful completion of the course, student will be able to*

CO1: Describe AI problems, foundations of AI, and agent architectures, and analyze how agents perceive, act, and make rational decisions in different environments (K2)

CO2: Apply uninformed and heuristic search strategies—including BFS, DFS, Hill Climbing, A*, AO*, and game-playing algorithms—to solve well-formulated problems (K3)

CO3: Develop suitable knowledge representation schemes using predicate logic, semantic networks, frames, rules, and probabilistic models to support reasoning under uncertainty (K3)

CO4: Perform inference using first-order logic, unification, resolution, forward/backward chaining, and apply learning approaches such as decision trees, EBL, statistical methods, and reinforcement learning (K4)

CO5: Examine the architecture, components, knowledge acquisition techniques, and shells of expert systems, and evaluate typical expert system applications (K4)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	-	-	-	-	-	-	-	-	L	-	-	-
CO2	H	H	H	M	L	M	-	-	-	-	-	L	-	-	-
CO3	H	M	H	M	M	M	-	-	-	L	-	L	-	-	-
CO4	H	H	H	H	M	M	-	-	-	L	-	M	-	-	-
CO5	M	M	M	M	H	L	-	-	-	L	-	M	-	-	-

UNIT - I

Introduction: AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

UNIT - II

Searching- Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A* ,AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem in Game playing, Alpha-Beta pruning, Evaluation functions.

UNIT - III

Representation of Knowledge: Knowledge representation issues, predicate logic- logic programming, semantic nets- frames and inheritance, constraint propagation, representing knowledge using rules, rules based deduction systems. Reasoning under uncertainty, review of probability, Bayes' probabilistic interferences and dempstershafer theory.

UNIT - IV

Logic concepts: First order logic. Inference in first order logic, propositional vs. first order inference, unification & lifts forward chaining, Backward chaining, Resolution, Learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods, Reinforcement Learning.

UNIT - V

Expert Systems: Architecture of expert systems, Roles of expert systems – Knowledge Acquisition Meta knowledge Heuristics. Typical expert systems – MYCIN, DART, XCON: Expert systems shells.

Textbooks:
1. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach", SecondEdition, Pearson Education.

2. Kevin Night and Elaine Rich, Nair B., "Artificial Intelligence (SIE)", Mc Graw Hill

Reference Books:
1. David Poole, Alan Mackworth, Randy Goebel,"Computational Intelligence: a logical approach", Oxford University Press.
2. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problemsolving", Fourth Edition, Pearson Education.
3. J. Nilsson, "Artificial Intelligence: A new Synthesis", Elsevier Publishers.
4. Artificial Intelligence, SarojKaushik, CENGAGE Learning.

Online Learning Resources:
1. https://ai.google/
2. https://swayam.gov.in/nd1_noc19_me71/preview

III Year I Semester	MICROPROCESSORS & MICROCONTROLLERS	L	T	P	C
		3	0	0	3

Course Objectives:

- To introduce fundamental architectural concepts of microprocessors and microcontrollers.
- To impart knowledge on addressing modes and instruction set of 8086 and 8051
- To introduce assembly language programming concepts
- To explain memory and I/O interfacing with 8086 and 8051
- To introduce16 bit and 32 bit microcontrollers.

Course Outcomes: *After successful completion of the course, student will be able to*

CO1: Explain the architecture, operational modes, timing, and interrupt mechanisms of the 8086 microprocessor (K2)

CO2: Apply addressing modes, instruction set, and assembler directives to develop assembly-level programs for the 8086 microprocessor (K3)

CO3: Design interfacing circuits for memories, I/O devices, interrupts, serial communication, and peripheral controllers commonly used with 8086 (K4)

CO4: Describe the architecture, SFRs, instruction set, and programming model of the 8051 microcontroller and develop basic assembly programs (K3)

CO5: Develop interface routines for timers, serial communication, interrupts, sensors, ADC/DAC, and external devices using 8051, and compare microprocessor/microcontroller families (K3)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	-	-	-	-	-	-	L	-	-	-	-	-
CO2	H	M	H	M	M	-	-	-	-	L	-	-	-	-	-
CO3	H	H	H	H	H	-	-	-	-	L	-	-	-	-	-
CO4	H	M	M	L	M	-	-	-	-	L	-	-	-	-	-
CO5	H	H	H	M	H	-	-	-	-	M	-	-	-	-	-

UNIT I:

8086 Architecture: Main features, pin diagram/description, 8086 microprocessor family, internal architecture, bus interfacing unit, execution unit, interrupts and interrupt response, 8086 system timing, minimum mode and maximum mode configuration.

UNIT II:

8086 Programming: Program development steps, instructions, addressing modes, assembler directives, writing simple programs with an assembler, assembly language program development tools.

UNIT III:

8086 Interfacing: Semiconductor memories interfacing (RAM, ROM), Intel 8255 programmable peripheral interface, Interfacing switches and LEDS, Interfacing seven segment displays, software and hardware interrupt applications, Intel 8251 USART architecture and interfacing, Intel 8237a DMA controller, stepper motor, A/D and D/A converters, Need for 8259 programmable interrupt controllers.

UNIT IV:

Microcontroller, Architecture of 8051, Special Function Registers(SFRs), I/O Pins Ports and Circuits, Instruction set, Addressing modes, Assembly language programming.

UNIT V:

Interfacing Microcontroller, Programming 8051 Timers, Serial Port Programming, Interrupts Programming, LCD & Keyboard Interfacing, ADC, DAC & Sensor Interfacing, External Memory Interface, Stepper Motor and Waveform generation, Comparison of Microprocessor, Microcontroller, PIC and ARM processors

Textbooks:

1. Microprocessors and Interfacing – Programming and Hardware by Douglas V Hall, SSSP Rao, Tata McGraw Hill Education Private Limited, 3rd Edition,1994.
2. K M Bhurchandi, A K Ray, Advanced Microprocessors and Peripherals, 3rd edition, McGraw Hill Education, 2017.
3. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, 2nd edition, Pearson, 2012.

Reference Books:

1. Ramesh S Gaonkar, Microprocessor Architecture Programming and Applications with the 8085, 6th edition, Penram International Publishing, 2013.
2. Kenneth J. Ayala, The 8051 Microcontroller, 3rd edition, Cengage Learning, 2004.

III Year I Semester	QUANTUM COMPUTING	L	T	P	C
		3	0	0	3

Pre-requisites:

Basic knowledge of Linear Algebra (vectors, matrices, eigenvalues, operations)

Fundamental concepts of Probability & Statistics

Basic understanding of Classical Computing (binary operations, logic gates, algorithmic thinking)

Introductory Physics knowledge (wave-particle concepts, basic mechanics/electromagnetics)

Basic Foundations of Programming (ability to understand algorithms and computational models)

Course Objectives:

To introduce the fundamentals of quantum computing, the problem-solving approach using finite dimensional mathematics

Course Outcomes: *After successful completion of the course, student will be able to*

CO1: Explain the evolution of quantum computing and distinguish between classical and quantum computing models, including bits vs qubits and logical operations (K2)

CO2: Apply foundational mathematics, physics, and biological concepts—including linear algebra, Hilbert spaces, superposition, entanglement, and genomics/proteomics—to analyze quantum computational principles (K3)

CO3: Demonstrate the representation, physical realization, and manipulation of qubits, and construct simple quantum circuits using single and multi-qubit gates (K3)

CO4: Analyze and implement foundational quantum algorithms such as Deutsch, Deutsch–Jozsa, Grover, and Shor, and compare their performance with classical algorithms (K4)

CO5: Examine noise sources, apply quantum error-correction and fault-tolerant mechanisms, and interpret modern applications of quantum information and cryptography (K4)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	-	-	-	-	-	-	L	-	L	-	-	-
CO2	H	H	H	M	L	M	-	-	-	L	-	L	-	-	-
CO3	H	H	H	M	M	M	-	-	-	L	-	L	-	-	-
CO4	H	H	H	H	M	M	-	-	-	M	-	M	-	-	-
CO5	M	H	M	M	L	L	-	-	-	M	-	M	-	-	-

UNIT - I

History of Quantum Computing: Importance of Mathematics, Physics and Biology. Introduction to Quantum Computing: Bits Vs Qubits, Classical Vs Quantum logical operations

UNIT - II

Background Mathematics: Basics of Linear Algebra, Hilbert space, Probabilities and measurements.

Background Physics: Paul's exclusion Principle, Superposition, Entanglement and super-symmetry, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis. Background Biology: Basic concepts of Genomics and Proteomics (Central Dogma)

UNIT - III

Qubit: Physical implementations of Qubit. Qubit as a quantum unit of information. The Bloch sphere Quantum Circuits: single qubit gates, multiple qubit gates, designing the quantum circuits. Bell states.

UNIT - IV

Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor's factorization algorithm, Grover's search algorithm.

UNIT - V

Noise and error correction: Graph states and codes, Quantum error correction, fault-tolerant computation. Quantum Information and Cryptography: Comparison between classical and quantum information theory. Quantum Cryptography, Quantum teleportation

Text Books:
1. Quantum Computation and Quantum Information, Nielsen M. A.,Cambridge
2. Programming Quantum Computers, Essential Algorithms and Code Samples, Eric R Johnson, Nic Harrigan, Mercedes Ginemo, Segovia, Oreilly

Reference Books:
1. Quantum Computing for Computer Scientists, Noson S. Yanofsk, Mirco A. Mannucci
2. Principles of Quantum Computation and Information,Benenti G., Casati G. and Strini G., Vol.I: Basic Concepts, Vol II
3. Basic Tools and Special Topics, World Scientific. Pittenger A. O., An Introduction to Quantum Computing Algorithms

III Year I Semester	DATA MINING LAB	L	T	P	C
		0	0	3	1.5

Pre-requisites: Data Base Management Systems, Python Programming

Course Objectives: The main objective of the course is to
- Inculcate Conceptual, Logical, and Physical design of Data Warehouses OLAP applications and OLAP deployment
- Design a data warehouse or data mart to present information needed by management in a form that is usable
- Emphasize hands-on experience working with all real data sets.
- Test real data sets using popular data mining tools such as WEKA, Python Libraries
- Develop ability to design various algorithms based on data mining tools.

Course Outcomes: *After successful completion of the course, student will be able to*

CO1: Construct data warehouses, design multidimensional models (Star/Snowflake/Fact Constellation), and perform ETL and OLAP operations using industrial tools (K3)

CO2: Explore and operate WEKA tools to load, preprocess, visualize datasets, and analyze attribute characteristics for machine learning tasks (K3)

CO3: Apply data preprocessing, association rule mining (Apriori/FP-Growth), and interpretation of rules to derive meaningful insights from datasets (K4)

CO4: Implement and evaluate classification and clustering algorithms (ID3, J48, NB, k-NN, k-means) using WEKA/Python/R, and compare model performance using ROC curves, entropy, SSE, and confusion matrices (K4)

CO5: Develop machine learning programs in Python/Java/R for association mining, Naive Bayes, clustering, chi-square computation, and visualization using Matplotlib, and interpret results (K4)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	H	M	H	-	-	-	-	L	-	-	-	-	-
CO2	H	M	M	M	H	-	-	-	-	L	-	-	-	-	-
CO3	H	H	H	H	H	-	-	-	-	L	-	-	-	-	-
CO4	H	H	H	H	H	-	-	-	L	M	-	-	-	-	-
CO5	H	H	H	H	H	-	-	-	L	M	-	-	-	-	-

Software Requirements: WEKA Tool/Python/R-Tool/Rapid Tool/Oracle Data mining
List of Experiments:

1. Creation of a Data Warehouse.
- Build Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration Tool, Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects, etc.,)

- Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc).
- Write ETL scripts and implement using data warehouse tools.
- Perform Various OLAP operations such slice, dice, roll up, drill up and pivot

2. Explore machine learning tool "WEKA"
- Explore WEKA Data Mining/Machine Learning Toolkit.
- Downloading and/or installation of WEKA data mining toolkit.
- Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.
- Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify panel, Cluster panel, Associate panel and Visualize panel)
- Study the arff file format Explore the available data sets in WEKA. Load a data set (ex. Weather dataset, Iris dataset, etc.)
- Load each dataset and observe the following:
    1. List the attribute names and they types
    2. Number of records in each dataset
    3. Identify the class attribute (if any)
    4. Plot Histogram
    5. Determine the number of records for each class.
    6. Visualize the data in various dimensions

3. Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets
- Explore various options available in Weka for preprocessing data and apply Unsupervised filters like Discretization, Resample filter, etc. on each dataset
- Load weather. nominal, Iris, Glass datasets into Weka and run Apriori Algorithm with different support and confidence values.
- Study the rules generated. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated.
- Derive interesting insights and observe the effect of discretization in the rule generation process.

4. Demonstrate performing classification on data sets Weka/R
- Load each dataset and run 1d3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.
- Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix.
- Load each dataset into Weka/R and perform Naïve-bayes classification and k-Nearest Neighbour classification. Interpret the results obtained.
- Plot RoC Curves

- Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.

5. Demonstrate performing clustering of data sets
   - Load each dataset into Weka/R and run simple k-means clustering algorithm with different values of k (number of desired clusters).
   - Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.
   - Explore other clustering techniques available in Weka/R.
   - Explore visualization features of Weka/R to visualize the clusters. Derive interesting insights and explain.

6. Demonstrate knowledge flow application on data sets into Weka/R
   - Develop a knowledge flow layout for finding strong association rules by using Apriori, FP Growth algorithms
   - Set up the knowledge flow to load an ARFF (batch mode) and perform a cross validation using J48 algorithm
   - Demonstrate plotting multiple ROC curves in the same plot window by using j48 and Random forest tree

7. Demonstrate ZeroR technique on Iris dataset (by using necessary preprocessing technique(s)) and share your observations
8. Write a java program to prepare a simulated data set with unique instances.
9. Write a Python program to generate frequent item sets / association rules using Apriori algorithm
10. Write a program to calculate chi-square value using Python/R. Report your observation.
11. Write a program of Naive Bayesian classification using Python/R programming language.
12. Implement a Java/R program to perform Apriori algorithm
13. Write a R program to cluster your choice of data using simple k-means algorithm using JDK
14. Write a program of cluster analysis using simple k-means algorithm Python/R programming language.
15. Write a program to compute/display dissimilarity matrix (for your own dataset containing at least four instances with two attributes) using Python
16. Visualize the datasets using matplotlib in python/R.(Histogram, Box plot, Bar chart, Pie chart etc.,)

III Year I Semester	**COMPUTER NETWORKS LAB**	**L**	**T**	**P**	**C**
		0	0	3	1.5

Course Objectives:
Learn basic concepts of computer networking and acquire practical notions of protocols with the emphasis on TCP/IP. A lab provides a practical approach to Ethernet/Internet networking: networks are assembled, and experiments are made to understand the layered architecture and how do some important protocols work

Course Outcomes: *After successful completion of the course, student will be able to*
CO1: Configure and operate basic network devices to establish Local Area Networks and analyze packet-level behavior using tools such as Wireshark (K4)
CO2: Implement data link layer framing, error detection and correction techniques such as bit/character stuffing, checksum, Hamming code, and CRC algorithms (K3)
CO3: Develop and simulate reliable data transfer protocols including Stop-and-Wait, Go-Back-N, and Selective Repeat using programming constructs (K3)
CO4: Implement routing algorithms such as Dijkstra's shortest path and Distance Vector routing to compute optimal routing paths (K4)
CO5: Simulate congestion control, packet flow behavior, and throughput using NS2/Nmap and evaluate network performance under various scenarios (K5)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	M	L	M	H	-	-	-	-	L	-	M	-	-	-
CO2	H	M	L	M	L	-	-	-	-	L	-	M	-	-	-
CO3	M	M	M	M	L	-	-	-	-	L	-	M	-	-	-
CO4	H	H	M	H	L	-	-	-	-	L	-	M	-	-	-
CO5	M	H	M	H	H	-	-	-	-	L	-	H	-	-	-

List of Experiments:
1. Study of Network devices in detail and connect the computers in Local Area Network.
2. Write a Program to implement the data link layer farming methods such as
   i) Character stuffing ii) bit stuffing.
3. Write a Program to implement data link layer farming method checksum.
4. Write a program for Hamming Code generation for error detection and correction.
5. Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.
6. Write a Program to implement Sliding window protocol for Goback N.
7. Write a Program to implement Sliding window protocol for Selective repeat.
8. Write a Program to implement Stop and Wait Protocol.
9. Write a program for congestion control using leaky bucket algorithm
10. Write a Program to implement Dijkstra's algorithm to compute the Shortest path through a graph.

11. Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).

12. Write a Program to implement Broadcast tree by taking subnet of hosts.

13. Wireshark

       i. Packet Capture Using Wire shark

       ii. Starting Wire shark

       iii. Viewing Captured Traffic

       iv. Analysis and Statistics & Filters.

14. How to run Nmap scan

15. Operating System Detection using Nmap

16. Do the following using NS2 Simulator

       i. NS2 Simulator-Introduction

       ii. Simulate to Find the Number of Packets Dropped

       iii. Simulate to Find the Number of Packets Dropped by TCP/UDP

       iv. Simulate to Find the Number of Packets Dropped due to Congestion

       v. Simulate to Compare Data Rate& Throughput.

III Year I Semester	FULL STACK DEVELOPMENT - 2	L	T	P	C
		0	1	2	2

Course Objectives:

The main objectives of the course are to

- Make use of router, template engine and authentication using sessions to develop application in Express JS.
- Build a single page application using RESTful APIs in Express JS
- Apply router and hooks in designing React JS application
- Make use of MongoDB queries to perform CRUD operations on document database

Course Outcomes: *After successful completion of the course, student will be able to*

CO1: Build server-side applications using ExpressJS with routing, middleware, templating, sessions, authentication, and RESTful APIs (K3)

CO2: Develop interactive client-side applications using ReactJS components, props, states, JSX, routing, forms, hooks, and event handling (K4)

CO3: Design and integrate MERN-based full-stack applications using ExpressJS, ReactJS, and MongoDB for end-to-end data management (K5)

CO4: Perform CRUD operations and manage collections, indexes, and aggregations using MongoDB and Mongoose (K3)

CO5: Implement and test real-world applications (e.g., To-Do, Quiz App) and evaluate security, performance, and usability considerations (K5)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	M	M	L	H	-	-	-	-	L	-	M	-	-	-
CO2	M	H	M	M	H	-	-	-	-	M	-	M	-	-	-
CO3	H	H	H	M	H	-	-	-	-	M	-	H	-	-	-
CO4	M	M	M	M	H	-	-	-	-	L	-	M	-	-	-
CO5	M	H	H	H	H	-	-	-	-	M	-	H	-	-	-

Experiments covering the Topics:

- Express JS – Routing, HTTP Methods, Middleware, Templating, Form Data
- Express JS – Cookies, Sessions, Authentication, Database, RESTful APIs
- React JS – Render HTML, JSX, Components – function & Class, Props and States, Styles, Respond to Events
- React JS – Conditional Rendering, Rendering Lists, React Forms, React Router, Updating the Screen
- React JS – Hooks, Sharing data between Components, Applications – To-do list and Quiz
- MongoDB – Installation, Configuration, CRUD operations, Databases, Collections and Records

Sample Experiments:

1. Express JS – Routing, HTTP Methods, Middleware.

a. Write a program to define a route, Handling Routes, Route Parameters, Query Parameters and URL building.

b. Write a program to accept data, retrieve data and delete a specified resource using http methods.

c. Write a program to show the working of middleware.

2. Express JS – Templating, Form Data

a. Write a program using templating engine.

b. Write a program to work with form data.

3. Express JS – Cookies, Sessions, Authentication

a. Write a program for session management using cookies and sessions.

b. Write a program for user authentication.

4. Express JS – Database, RESTful APIs

a. Write a program to connect MongoDB database using Mangoose and perform CRUD operations.

b. Write a program to develop a single page application using RESTful APIs.

5. ReactJS – Render HTML, JSX, Components – function & Class

a. Write a program to render HTML to a web page.

b. Write a program for writing markup with JSX.

c. Write a program for creating and nesting components (function and class)

6. ReactJS – Props and States, Styles, Respond to Events

a. Write a program to work with props and states.

b. Write a program to add styles (CSS & Sass Styling) and display data.

c. Write a program for responding to events.

7. ReactJS – Conditional Rendering, Rendering Lists, React Forms

a. Write a program for conditional rendering.

b. Write a program for rendering lists.

c. Write a program for working with different form fields using react forms.

8. ReactJS – React Router, Updating the Screen

a. Write a program for routing to different pages using react router.

b. Write a program for updating the screen.

9. ReactJS – Hooks, Sharing data between Components

a. Write a program to understand the importance of using hooks.

b.   Write a program for sharing data between components.

10. MongoDB – Installation, Configuration, CRUD operations
a.   Install MongoDB and configure ATLAS
b.   Write MongoDB queries to perform CRUD operations on document using insert(), find(), update(), remove()

11. MongoDB – Databases, Collections and Records
a.   Write MongoDB queries to Create and drop databases and collections.
b.   Write MongoDB queries to work with records using find(), limit(), sort(), createIndex(), aggregate().

12. Augmented Programs: (Any 2 must be completed)
a.   Design a to-do list application using NodeJS and ExpressJS.
b.   Design a Quiz app using ReactJS.
c.   Complete the MongoDB certification from MongoDB University website.

Text Books:
1. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.
2. Node.Js in Action, Mike Cantelon, Mark Harter, T.J. Holowaychuk, Nathan Rajlich,Manning Publications. (Chapters 1-11)
3. React Quickly, AzatMardan,Manning Publications(Chapters 1-8,12-14)

Web Links:
1. ExpressJS - https://www.tutorialspoint.com/expressjs
2. ReactJS - https://www.w3schools.com/REACT (and)   https://react.dev/learn#
3. MongoDB - https://learn.mongodb.com/learning-paths/introduction-to-mongodb

III Year I Semester	USER INTERFACE DESIGN USING FLUTTER	L	T	P	C
		0	0	2	1

Course Objectives:
- Learns to Implement Flutter Widgets and Layouts
- Understands Responsive UI Design and with Navigation in Flutter
- Knowledge on Widges and customize widgets for specific UI elements, Themes
- Understand to include animation apart from fetching data

Course Outcomes: *After successful completion of the course, student will be able to*
CO1: Apply Dart programming constructs and core Flutter widgets to build basic user interfaces (K3)
CO2: Develop responsive and multi-screen mobile applications using layouts, navigation, and routing techniques (K4)
CO3: Implement effective state management and custom widget design using Flutter frameworks such as setState and Provider (K4)
CO4: Integrate animations, form handling, validation, and REST API communication to build interactive and data-driven mobile applications (K5)
CO5: Perform debugging, testing, and performance optimization of Flutter applications using available tools, testing frameworks, and best practices (K5)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	L	M	L	H	-	-	-	-	L	-	M	-	-	-
CO2	M	M	H	L	H	-	-	-	-	M	-	M	-	-	-
CO3	M	H	H	L	H	-	-	-	-	M	-	H	-	-	-
CO4	M	H	H	M	H	-	-	-	-	M	-	H	-	-	-
CO5	M	H	M	H	H	-	-	-	-	M	-	H	-	-	-

List of Experiments:
Students need to implement the following experiments

1. a) Install Flutter and Dart SDK.
   b) Write a simple Dart program to understand the language basics.
2. a) Explore various Flutter widgets (Text, Image, Container, etc.).
   b) Implement different layout structures using Row, Column, and Stack widgets.
3. a) Design a responsive UI that adapts to different screen sizes.
   b) Implement media queries and breakpoints for responsiveness.
4. a) Set up navigation between different screens using Navigator.
   b) Implement navigation with named routes.
5. a) Learn about stateful and stateless widgets.
   b) Implement state management using set State and Provider.
6. a) Create custom widgets for specific UI elements.
   b) Apply styling using themes and custom styles.

7. a) Design a form with various input fields.
      b) Implement form validation and error handling.
8. a) Add animations to UI elements using Flutter's animation framework.
      b) Experiment with different types of animations (fade, slide, etc.).
9. a) Fetch data from a REST API.
      b) Display the fetched data in a meaningful way in the UI.
10. a) Write unit tests for UI components.
      b) Use Flutter's debugging tools to identify and fix issues.

Text Books:
1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.
2. Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1st Edition, Apres
3. Richard Rose, Flutter & Dart Cookbook, Developing Full stack Applications for the Cloud, Oreilly.

III Year II Semester	COMPILER DESIGN	L	T	P	C
		3	0	0	3

Course Objectives:

Understand the basic concept of compiler design, and its different phases which will be helpful to construct new tools like LEX, YACC, etc.

Course Outcomes: *After successful completion of the course, student will be able to*
CO1: Explain lexical and syntax analysis mechanisms, token specification, and grammar properties used in compiler front-end design (K2)
CO2: Apply top-down and bottom-up parsing techniques to analyze context-free grammars and construct predictive and LR parsing tables (K3)
CO3: Develop syntax-directed definitions, translation schemes, and intermediate code representations such as TAC, syntax trees, and backpatching (K4)
CO4: Examine code optimization techniques, data-flow analysis, and loop optimizations to improve intermediate code efficiency (K4)
CO5: Construct runtime environments, activation records, and generate object code by addressing issues in code generation and register allocation (K5)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	L	L	-	-	-	-	L	-	M	-	-	-
CO2	H	H	M	L	M	-	-	-	-	L	-	M	-	-	-
CO3	H	H	H	M	H	-	-	-	-	L	-	H	-	-	-
CO4	M	H	M	H	H	-	-	-	-	L	-	H	-	-	-
CO5	H	H	H	M	H	-	-	-	-	L	-	H	-	-	-

UNIT I:
Lexical Analysis: Language Processors, Structure of a Compiler, Lexical Analysis, The Role of the Lexical Analyzer, Bootstrapping, Input Buffering, Specification of Tokens, Recognition of Tokens, Lexical Analyzer Generator-LEX, Finite Automata, Regular Expressions and Finite Automata, Design of a Lexical Analyzer Generator.
Syntax Analysis: The Role of the Parser, Context-Free Grammars, Derivations, Parse Trees, Ambiguity, Left Recursion, Left Factoring,

UNIT II:
Top Down Parsing: Pre Processing Steps of Top Down Parsing, Backtracking, Recursive Descent Parsing, LL (1) Grammars, Non-recursive Predictive Parsing, Error Recovery in Predictive Parsing.
Bottom Up Parsing: Introduction, Difference between LR and LL Parsers, Types of LR Parsers, Shift Reduce Parsing, SLR Parsers, Construction of SLR Parsing Tables, More Powerful LR Parses, Construction of CLR (1) and LALR Parsing Tables, Dangling Else Ambiguity, Error Recovery in LR Parsing, Handling Ambiguity Grammar with LR Parsers.

UNIT III:
Syntax Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax Directed Translation, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's. **Intermediate Code Generation:** Variants of Syntax Trees, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Intermediate Code for Procedures.

UNIT IV:
Code Optimization: The Principle Sources of Optimization, Basic Blocks, Optimization of Basic Blocks, Structure Preserving Transformations, Flow Graphs, Loop Optimization, Data-Flow Analysis, Peephole Optimization

UNIT V:
Run Time Environments: Storage Organization, Run Time Storage Allocation, Activation Records, Procedure Calls, Displays
Code Generation: Issues in the Design of a Code Generator, Object Code Forms, Code Generation Algorithm, Register Allocation and Assignment.

Text Books:
1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, Pearson, 2007.

Reference Books:
1. Compiler Construction, Principles and Practice, Kenneth C Louden, Cengage Learning, 2006
2. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
3. Optimizing Compilers for Modern Architectures, Randy Allen, Ken Kennedy, Morgan Kauffmann, 2001.
4. Levine, J.R., T. Mason and D. Brown, Lex and Yacc, edition, O'Reilly & Associates, 1990

III Year II Semester	CLOUD COMPUTING	L	T	P	C
		3	0	0	3

Course Objectives:
- To explain the evolving utility computing model called cloud computing.
- To introduce the various levels of services offered by cloud.
- To discuss the fundamentals of cloud enabling technologies such as distributed computing, service-oriented architecture and virtualization.
- To emphasize the security and other challenges in cloud computing.
- To introduce the advanced concepts such as containers, serverless computing and cloud-centric Internet of Things.

Course Outcomes: *After successful completion of the course, student will be able to*

CO1: Explain cloud computing concepts, service models, deployment models, and associated benefits in modern distributed systems (K2)

CO2: Apply cloud-enabling technologies such as parallel computing, distributed systems, SOA, and virtualization in cloud environments (K3)

CO3: Analyze virtualization techniques, container technologies, orchestration tools, and public cloud VM/container services (K4)

CO4: Evaluate cloud computing challenges related to interoperability, scalability, energy efficiency, and cloud security architecture (K5)

CO5: Assess advanced cloud paradigms including serverless computing, cloud-centric IoT, edge/fog computing, DevOps, and quantum cloud technologies (K5)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	L	L	-	L	-	-	L	-	M	-	-	-
CO2	H	H	M	M	M	-	L	-	-	L	-	M	-	-	-
CO3	H	H	H	M	H	-	L	-	-	L	-	H	-	-	-
CO4	H	H	M	H	H	-	M	L	-	L	-	H	-	-	-
CO5	H	H	M	H	H	-	M	L	-	L	-	H	-	-	-

UNIT -I: Introduction to Cloud Computing Fundamentals
Cloud computing at a glance, defining a cloud, cloud computing reference model, types of services (IaaS, PaaS, SaaS), cloud deployment models (public, private, hybrid), utility computing, cloud computing characteristics and benefits, cloud service providers (Amazon Web Services, Microsoft Azure, Google AppEngine).

UNIT-II: Cloud Enabling Technologies
Ubiquitous Internet, parallel and distributed computing, elements of parallel computing, hardware architectures for parallel computing (SISD, SIMD, MISD, MIMD), elements of distributed computing, Inter-process communication, technologies for distributed computing, remote procedure calls (RPC), service-oriented architecture (SOA), Web services, virtualization.

UNIT-III: Virtualization and Containers

Characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud Computing, pros and cons of virtualization, technology examples (XEN, VMware), building blocks of containers, container platforms (LXC, Docker), container orchestration, Docker Swarm and Kubernetes, public cloud VM (e.g. Amazon EC2) and container (e.g. Amazon Elastic Container Service) offerings.

UNIT-IV: Cloud computing challenges

Economics of the cloud, cloud interoperability and standards, scalability and fault tolerance, energy efficiency in clouds, federated clouds, cloud computing security, fundamentals of computer security, cloud security architecture, cloud shared responsibility model, security in cloud deployment models.

UNIT -V: Advanced concepts in cloud computing

Serverless computing, Function-as-a-Service, serverless computing architecture, public cloud (e.g. AWS Lambda) and open-source (e.g. OpenFaaS) serverless platforms, Internet of Things (IoT), applications, cloud-centric IoT and layers, edge and fog computing, DevOps, infrastructure-as-code, quantum cloud computing.

Text Books:

1. Mastering Cloud Computing, 2nd edition, Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi, Shivananda Poojara, Satish N. Srirama, Mc Graw Hill, 2024.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.

Reference Books:

1. Cloud Computing, Theory and Practice, Dan C Marinescu, 2nd edition, MK Elsevier, 2018.
2. Essentials of cloud Computing, K. Chandrasekhran, CRC press, 2014.
3. Online documentation and tutorials from cloud service providers (e.g., AWS, Azure, GCP)

III Year II Semester	CRYPTOGRAPHY & NETWORK SECURITY (Common to CSE, CS & IT)	L	T	P	C
		3	0	0	3

Course Objectives:
The main objectives of this course are to explore the working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, public key algorithms, design issues and working principles of various authentication protocols and various secure communication standards including Kerberos, IPsec, and SSL/TLS.

UNIT I:
Basic Principles: Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography- integer arithmetic, modular arithmetic, matrices, linear conguence.

UNIT II:
Symmetric Encryption: Mathematics of Symmetric Key Cryptography-algebraic structures, $GF(2^n)$ Fields, Introduction to Modern Symmetric Key Ciphers-modern block ciphers, modern stream ciphers, Data Encryption Standard- DES structure, DES analysis, Security of DES, Multiple DES, Advanced Encryption Standard-transformations, key expansions, AES ciphers, Analysis of AES

UNIT III:
Asymmetric Encryption: Mathematics of Asymmetric Key Cryptography-primes, primality testing, factorization, CRT, Asymmetric Key Cryptography- RSA crypto system, Rabin cryptosystem, Elgamal Crypto system, ECC

UNIT IV:
Data Integrity, Digital Signature Schemes & Key Management: Message Integrity and Message Authentication-message integrity, Random Oracle model, Message authentication, Cryptographic Hash Functions-whirlpool, SHA-512, Digital Signature- process, services, attacks, schemes, applications, Key Management-symmetric key distribution, Kerberos.

UNIT V:
Network Security-I: Security at application layer: PGP and S/MIME, Security at the Transport Layer: SSL and TLS, **Network Security-II :** Security at the Network Layer: IPSec-two modes, two security protocols, security association, IKE, ISAKMP, System Security-users, trust, trusted systems, buffer overflow, malicious software, worms, viruses, IDS, Firewalls.

Text Books:

1. Cryptography and Network Security, 3rd Edition Behrouz A Forouzan, Deb deep Mukhopadhyay, McGraw Hill,2015
2. Cryptography and Network Security,4th Edition, William Stallings, (6e) Pearson,2006
3. Everyday Cryptography, 1st Edition, Keith M.Martin, Oxford, 2016

Reference Books:

1. Network Security and Cryptography, 1st Edition, Bernard Meneges, Cengage Learning, 2018

III Year II Semester	**SOFTWARE TESTING METHODOLOGIES** (Common to CSE, CS, IT, CSD, CSE(AI), CSE (AI&ML), CSE (AI&DS))	L	T	P	C
		3	0	0	3

Course Objectives

- To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.
- To develop skills in software test automation and management using the latest tools.

UNIT - I

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II

Transaction Flow Testing: transaction flows, transaction flow testing techniques.

Data Flow testing: Basics of data flow testing, strategies in data flow testing, application of data flow testing.

Domain Testing: domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

UNIT - III

Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

UNIT - IV

State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips.

UNIT - V

Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like Jmeter/selenium/soapUI/Catalon).

Text Books:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

Reference Books:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

III Year II Semester	CYBER SECURITY	L	T	P	C
		3	0	0	3

Course Objectives:
The aim of the course is to
- identify security risks and take preventive steps
- understand the forensics fundamentals
- understand the evidence capturing process
- understand the preservation of digital evidence

UNIT I: Introduction to Cybercrime: Introduction, Cybercrime: Definition and Origins of the Word, Cybercrime and Information Security, Cyber criminals, Classifications of Cybercrime, Cyberstalking, Cybercafe and Cybercrimes, Botnets. Attack Vector, Proliferation of Mobile and Wireless Devices, Security Challenges Posed by Mobile Devices, Attacks on Mobile/Cell Phones, Network and Computer Attacks.

UNIT II: Tools and Methods : Proxy Servers and Anonymizers, Phishing, Password Cracking, Keyloggers and Spywares, Virus and Worms, Trojan Horses and Backdoors, Steganography, Sniffers, Spoofing, Session Hijacking Buffer over flow, DoS and DDoS Attacks, SQL Injection, Buffer Overflow, Attacks on Wireless Networks, Identity Theft (ID Theft), Foot Printing and Social Engineering, Port Scanning, Enumeration.

UNIT III: Cyber Crime Investigation: Introduction, Investigation Tools, eDiscovery, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E-Mail Recovery, Hands on Case Studies. Encryption and Decryption Methods, Search and Seizure of Computers, Recovering Deleted Evidences, Password Cracking.

UNIT IV: Computer Forensics and Investigations: Understanding Computer Forensics, Preparing for Computer Investigations. Current Computer Forensics Tools: Evaluating Computer Forensics Tools, Computer Forensics Software Tools, Computer Forensics Hardware Tools, Validating and Testing Forensics Software, Face, Iris and Fingerprint Recognition, Audio Video Analysis, Windows System Forensics, Linux System Forensics, Graphics and Network Forensics, E-mail Investigations, Cell Phone and Mobile Device Forensics.

UNIT V: Cyber Crime Legal Perspectives: Introduction, Cybercrime and the Legal Landscape around the World, The Indian IT Act, Challenges to Indian Law and Cybercrime Scenario in India, Consequences of Not Addressing the Weakness in Information Technology Act, Digital Signatures and the Indian IT Act, Amendments to the Indian IT Act, Cybercrime andPunishment, Cyberlaw, Technology and Students: Indian Scenario.

Text Books:
1. Sunit Belapure Nina Godbole "Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", WILEY, 2011.
2. Nelson Phillips and Enfinger Steuart, "Computer Forensics and Investigations", Cengage Learning, New Delhi, 2009.

Reference Books:
1. Michael T. Simpson, Kent Backman and James E. Corley, "Hands on Ethical Hacking and Network Defence", Cengage, 2019.
2. Computer Forensics, Computer Crime Investigation by John R. Vacca, Firewall Media, New Delhi.
3. Alfred Basta, Nadine Basta,Mary Brown and Ravinder Kumar "Cyber Security and Cyber Laws" , Cengage,2018.

E-Resources:
1. CERT-In Guidelines- http://www.cert-in.org.in/
2. https://www.coursera.org/learn/introduction-cybersecurity-cyber-attacks [Online Course]
3. https://computersecurity.stanford.edu/free-online-videos [Free Online Videos]
4. Nickolai Zeldovich. 6.858 Computer Systems Security. Fall 2014. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu License: Creative Commons BY-NC-SA.

III Year II Semester	**DevOps** **(Common to CSE, CS, IT, AI &Ml, CSE (AI), CSE AI&ML)**	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of this course are to:

- Describe the agile relationship between development and IT operations.
- Understand the skill sets and high-functioning teams involved in DevOps and related methods to reach a continuous delivery capability.
- Implement automated system update and DevOps lifecycle.

UNIT-I

Introduction to DevOps: Introduction to SDLC, Agile Model. Introduction to Devops. DevOps Features, DevOps Architecture, DevOps Lifecycle, Understanding Workflow and principles, Introduction to DevOps tools, Build Automation, Delivery Automation, Understanding Code Quality, Automation of CI/ CD. Release management, Scrum, Kanban, delivery pipeline, bottlenecks, examples

UNIT-II

Source Code Management (GIT): The need for source code control, The history of source code management, Roles and code, source code management system and migrations. What is Version Control and GIT, GIT Installation, GIT features, GIT workflow, working with remote repository, GIT commands, GIT branching, GIT staging and collaboration. UNIT TESTING - CODE COVERAGE: Junit, nUnit & Code Coverage with Sonar Qube, SonarQube - Code Quality Analysis.

UNIT-III

Build Automation - Continuous Integration (CI): Build Automation, What is CI Why Cl is Required, CI tools, Introduction to Jenkins (With Architecture), jenkins workflow, jenkins master slave architecture, Jenkins Pipelines, PIPELINE BASICS - Jenkins Master, Node, Agent, and Executor Freestyle Projects & Pipelines, Jenkins for Continuous Integration, Create and Manage Builds, User Management in Jenkins Schedule Builds, Launch Builds on Slave Nodes.

UNIT-IV

Continuous Delivery (CD): Importance of Continuous Delivery, CONTINUOUS DEPLOYMENT CD Flow, Containerization with Docker: Introduction to Docker, Docker installation, Docker commands, Images & Containers, DockerFile, Running containers, Working with containers and publish to Docker Hub.

Testing Tools: Introduction to Selenium and its features, JavaScript testing.

UNIT-V

Configuration Management - ANSIBLE: Introduction to Ansible, Ansible tasks, Roles, Jinja templating, Vaults, Deployments using Ansible.

CONTAINERIZATION USING KUBERNETES(OPENSHIFT): Introduction to Kubernetes Namespace & Resources, CI/CD - On OCP, BC, DC & ConfigMaps, Deploying Apps on Openshift Container Pods. Introduction to Puppet master and Chef.

Text Books:

1. Joyner, Joseph., Devops for Beginners: Devops Software Development Method Guide for Software Developers and It Professionals, 1st Edition Mihails Konoplows, 2015.
2. Alisson Machado de Menezes., Hands-on DevOps with Linux,1st Edition, BPB Publications, India, 2021.

Reference Books:

1. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley; ISBN-10
2. Gene Kim Je Humble, Patrick Debois, John Willis. The DevOps Handbook, 1st Edition, IT Revolution Press, 2016.
3. Verona, Joakim Practical DevOps, 1st Edition, Packt Publishing, 2016.
4. Joakim Verona. Practical Devops, Ingram short title; 2nd edition (2018). ISBN10: 1788392574
5. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications. ISBN: 9788126579952

III Year II Semester	MACHINE LEARNING	L	T	P	C
		3	0	0	3

Course Objectives:

The objectives of the course is to

- Define machine learning and its different types (supervised and unsupervised) and understand their applications.
- Apply supervised learning algorithms including decision trees and k-nearest neighbours (k-NN).
- Implement unsupervised learning techniques, such as K-means clustering.

UNIT-I: Introduction to Machine Learning: Evolution of Machine Learning, Paradigms for ML, Learning by Rote, Learning by Induction, Reinforcement Learning, Types of Data, Matching, Stages in Machine Learning, Data Acquisition, Feature Engineering, Data Representation, Model Selection, Model Learning, Model Evaluation, Model Prediction, Search and Learning, Data Sets.

UNIT-II: Nearest Neighbor-Based Models: Introduction to Proximity Measures, Distance Measures, Non-Metric Similarity Functions, Proximity Between Binary Patterns, Different Classification Algorithms Based on the Distance Measures ,K-Nearest Neighbor Classifier, Radius Distance Nearest Neighbor Algorithm, KNN Regression, Performance of Classifiers, Performance of Regression Algorithms.

UNIT-III: Models Based on Decision Trees: Decision Trees for Classification, Impurity Measures, Properties, Regression Based on Decision Trees, Bias–Variance Trade-off, Random Forests for Classification and Regression. The Bayes Classifier: Introduction to the Bayes Classifier, Bayes' Rule and Inference, The Bayes Classifier and its Optimality, Multi-Class Classification, Class Conditional Independence and Naive Bayes Classifier (NBC)

UNIT-IV: Linear Discriminants for Machine Learning: Introduction to Linear Discriminants, Linear Discriminants for Classification, Perceptron Classifier, Perceptron Learning Algorithm, Support Vector Machines, Linearly Non-Separable Case, Non-linear SVM, Kernel Trick, Logistic Regression, Linear Regression, Multi-Layer Perceptrons (MLPs), Backpropagation for Training an MLP.

UNIT-V: Clustering : Introduction to Clustering, Partitioning of Data, Matrix Factorization, Clustering of Patterns, Divisive Clustering, Agglomerative Clustering, Partitional Clustering, K-Means Clustering, Soft Partitioning, Soft Clustering, Fuzzy C-Means Clustering, Rough Clustering, Rough K-Means Clustering Algorithm, Expectation Maximization-Based Clustering, Spectral Clustering.

Text Books:

1. "Machine Learning Theory and Practice", M N Murthy, V S Ananthanarayana, Universities Press (India), 2024

Reference Books:

1. "Machine Learning", Tom M. Mitchell, McGraw-Hill Publication, 2017

2. "Machine Learning in Action",Peter Harrington, DreamTech

3. "Introduction to Data Mining", Pang-Ning Tan, Michel Stenbach, Vipin Kumar, 7th Edition, 2019.

III Year II Semester	SOFTWARE PROJECT MANAGEMENT	L	T	P	C
		3	0	0	3

Course Objectives:

At the end of the course, the student shall be able to:

- To describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project
- To compare and differentiate organization structures and project structures
- To implement a project to manage project schedule, expenses and resources with the application of suitable project management tools

UNIT-I:

Conventional Software Management: The waterfall model, conventional software Management performance.

Evolution of Software Economics: Software Economics, pragmatic software cost estimation.

Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.

The old way and the new: The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

UNIT-II:

Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases.

Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.

UNIT- III:

Model based software architectures: A Management perspective and technical perspective.

Work Flows of the process: Software process workflows, Iteration workflows.

Checkpoints of the process: Major mile stones, Minor Milestones, Periodic status assessments.

Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.

UNIT- IV:

Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.

Process Automation: Automation Building blocks, The Project Environment.

Project Control and Process instrumentation: The seven core Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation.

UNIT-V:
Agile Methodology, ADAPTing to Scrum, Patterns for Adopting Scrum, Iterating towards Agility. **Fundamentals of DevOps**: Architecture, Deployments, Orchestration, Need, Instance of applications, DevOps delivery pipeline, DevOps eco system. DevOps adoption in projects: Technology aspects, Agiling capabilities, Tool stack implementation, People aspect, processes

Text Books:
1. Software Project Management, Walker Royce, PEA, 2005.
2. Succeeding with Agile: Software Development Using Scrum, Mike Cohn, Addison Wesley.
3. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, Gene Kim , John Willis , Patrick Debois , Jez Humb,1st Edition, O'Reilly publications, 2016.

Reference Books:
1. Software Project Management, Bob Hughes,3/e, Mike Cotterell, TMH
2. Software Project Management, Joel Henry, PEA
3. Software Project Management in practice, Pankaj Jalote, PEA, 2005,
4. Effective Software Project Management, Robert K.Wysocki, Wiley,2006.
5. Project Management in IT, Kathy Schwalbe, Cengage

		L	T	P	C
III Year II Semester	**MOBILE ADHOC NETWORKS (Common to CSE, CS, IT, CSE (AI), CSE (AI &ML, CSD)**	3	0	0	3

Course Objectives:

From the course the student will learn

- Architect sensor networks for various application setups.
- Devise appropriate data dissemination protocols and model links cost.
- Understanding of the fundamental concepts of wireless sensor networks and has a basic knowledge of the various protocols at various layers.
- Evaluate the performance of sensor networks and identify bottlenecks.

UNIT I: Introduction to Ad Hoc Wireless Networks- Cellular and Ad Hoc Wireless Networks, Characteristics of MANETs, Applications of MANETs, Issues and Challenges of MANETs, Ad Hoc Wireless Internet, MAC protocols for Ad hoc Wireless Networks-Issues, Design Goals and Classifications of the MAC Protocols.

UNIT II: Routing Protocols for Ad Hoc Wireless Networks- Issues in Designing a Routing Protocol, Classifications of Routing Protocols, Topology-based versus Position-based Approaches, Issues and design goals of a Transport layer protocol, Classification of Transport layer solutions, TCP over Ad hoc Wireless Networks, Solutions for TCP over Ad Hoc Wireless Networks, Other Transport layer protocols.

UNIT III: Security protocols for Ad hoc Wireless Networks- Security in Ad hoc Wireless Networks, Network Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Key Management, Secure Routing in Ad hoc Wireless Networks, Cooperation in MANETs, Intrusion Detection Systems.

UNIT IV: Basics of Wireless Sensors and Applications- The Mica Mote, Sensing and Communication Range, Design Issues, Energy Consumption, Clustering of Sensors, Applications, Data Retrieval in Sensor Networks-Classification of WSNs, MAC layer, Routing layer, Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs.

UNIT V: Security in WSNs- Security in WSNs, Key Management in WSNs, Secure Data Aggregation in WSNs, Sensor Network Hardware-Components of Sensor Mote, Sensor Network Operating Systems–TinyOS, LA-TinyOS, SOS, RETOS, Imperative Language-nesC, **Dataflow Style Language-**TinyGALS, Node-Level Simulators, NS-2 and its sensor network extension, TOSSIM.

Text Books:

1. Ad Hoc Wireless Networks – Architectures and Protocols, 1st edition, C. Siva Ram Murthy, B. S. Murthy, Pearson Education, 2004
2. Ad Hoc and Sensor Networks – Theory and Applications, 2nd edition *Carlos Corderio Dharma P.Aggarwal,* World Scientific Publications / Cambridge University Press, March 2006

Reference Books:

1. Wireless Sensor Networks: An Information Processing Approach, 1st edition*, Feng Zhao, Leonidas Guibas*, Elsevier Science imprint, Morgan Kauffman Publishers, 2005, rp2009
2. Wireless Ad hoc Mobile Wireless Networks – Principles, Protocols and Applications, 1st edition*,* Subir Kumar Sarkar, et al., Auerbach Publications, Taylor & Francis Group, 2008
3. Ad hoc Networking, 1st edition*,Charles E.Perkins*, Pearson Education, 2001
4. Wireless Ad hoc Networking, 1st edition*, Shih-Lin Wu, Yu-Chee Tseng,* Auerbach Publications, Taylor & Francis Group, 2007
5. Wireless Sensor Networks – Principles and Practice, 1st edition, Fei Hu, Xiaojun Cao, An Auerbach book, CRC Press, Taylor & Francis Group, 2010

III Year II Semester	**NATURAL LANGUAGE PROCESSING**	L	T	P	C
		3	0	0	3

Course Objectives:

This course introduces the fundamental concepts and techniques of natural language processing (NLP).

- Students will gain an in-depth understanding of the computational properties of natural languages and the commonly used algorithms for processing linguistic information.
- The course examines NLP models and algorithms using both the traditional symbolic and the more recent statistical approaches.
- Enable students to be capable to describe the application based on natural language processing and to show the points of syntactic, semantic and pragmatic processing.

UNIT I:

INTRODUCTION: Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance.

UNIT II:

WORD LEVEL ANALYSIS: Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part- of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

UNIT III:

SYNTACTIC ANALYSIS: Context-Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures

UNIT IV:

SEMANTICS AND PRAGMATICS: Requirements for representation, First-Order Logic, Description Logics – Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods.

UNIT V:

DISCOURSE ANALYSIS AND LEXICAL RESOURCES: Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution using Hobbs and Centering

Algorithm – Coreference Resolution – Resources: Porter Stemmer, Lemmatizer, Penn Treebank, Brill's Tagger, WordNet, PropBank, FrameNet, Brown Corpus, British National Corpus (BNC).

Text Books:
1. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, 2ndEdition, Daniel Jurafsky, James H. Martin - Pearson Publication,2014.
2. Natural Language Processing with Python, First Edition, Steven Bird, Ewan Klein and Edward Loper, OReilly Media,2009.

Reference Books:
1. Language Processing with Java and Ling Pipe Cookbook, 1stEdition, Breck Baldwin, Atlantic Publisher, 2015.
2. Natural Language Processing with Java, 2ndEdition, Richard M Reese, OReilly Media,2015.
3. Handbook of Natural Language Processing, Second, NitinIndurkhya and Fred J. Damerau, Chapman and Hall/CRC Press, 2010.Edition
4. Natural Language Processing and Information Retrieval, 3rdEdition, TanveerSiddiqui, U.S. Tiwary, Oxford University Press,2008.

III Year II Semester	BIG DATA ANALYTICS	L	T	P	C
		3	0	0	3

Course Objectives: This course is aimed at enabling the students to

- To provide an overview of an exciting growing field of big data analytics.
- To introduce the tools required to manage and analyze big data like Hadoop, NoSQL, Map Reduce, HIVE, Cassandra, Spark.
- To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
- To optimize business decisions and create competitive advantage with Big Data analytics

UNIT I: big data, convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data, credit risk management, big data and algorithmic trading, big data and healthcare, big data in medicine, advertising and big data, big data technologies, introduction to Hadoop, open source technologies, cloud and big data, mobile business intelligence, Crowd sourcing analytics, inter and trans firewall analytics.

UNIT II: Introduction to NoSQL, aggregate data models, aggregates, key-value and document data models, relationships, graph databases, schema less databases, materialized views, distribution models, sharding, master-slave replication, peer- peer replication, sharding and replication, consistency, relaxing consistency, version stamps, Working with Cassandra ,Table creation, loading and reading data.

UNIT III: Data formats, analyzing data with Hadoop, scaling out, Architecture of Hadoop distributed file system (HDFS), fault tolerance, with data replication, High availability, Data locality , Map Reduce Architecture, Process flow, Java interface, data flow, Hadoop I/O, data integrity, compression, serialization. Introduction to Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation, Logical joins, Window functions, Optimization, Table partitioning, Bucketing, Indexing, Join strategies.

UNIT IV: Apache spark- Advantages over Hadoop, lazy evaluation, In memory processing, DAG, Spark context, Spark Session, RDD, Transformations- Narrow and Wide, Actions, Data frames ,RDD to Data frames, Catalyst optimizer, Data Frame Transformations, Working with Dates and Timestamps, Working with Nulls in Data, Working with Complex Types, Working with JSON, Grouping, Window Functions, Joins, Data Sources, Broadcast Variables, Accumulators, Deploying Spark- On-Premises Cluster Deployments, Cluster Managers- Standalone Mode, Spark on YARN , Spark Logs, The Spark UI- Spark UI History Server, Debugging and Spark First Aid

UNIT V: Spark-Performance Tuning, Stream Processing Fundamentals, Event-Time and State full Processing - Event Time, State full Processing, Windows on Event Time- Tumbling Windows, Handling Late Data with Watermarks, Dropping Duplicates in a Stream, Structured Streaming Basics - Core Concepts, Structured Streaming in Action, Transformations on Streams, Input and Output.

Text Books:
1. Big Data, Big Analytics: Emerging, Michael Minnelli, Michelle Chambers, and AmbigaDhiraj, 1st edition ,2013
2. SPARK: The Definitive Guide, Bill Chambers &MateiZaharia, O'Reilley, 2018-first Edition.
3. Business Intelligence and Analytic Trends for Today's Businesses", Wiley, First edition-2013.
4. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World Polyglot Persistence", Addison-Wesley Professional, 2012
5. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012

Reference Books:
1. "Hadoop Operations", O'Reilley, Eric Sammer, First Edition -2012.
2. "Programming Hive", O'Reilley, E. Capriolo, D. Wampler, and J. Rutherglen, 2012.
3. "HBase: The Definitive Guide", O'Reilley, Lars George, September 2011: First Edition..
4. "Cassandra: The Definitive Guide", O'Reilley, Eben Hewitt, 2010.
"Programming Pig", O'Reilley, Alan Gates, October 2011: First Edition

III Year II Semester	DISTRIBUTED OPERATING SYSTEM	L	T	P	C
		3	0	0	3

Course Objectives:

The main objective of the course is to introduce design issues and different message passing techniques in DOS, distributed systems, RPC implementation and its performance in DOS, distributed shared memory and resource management, distributed file systems and evaluate the performance in terms of fault tolerance, file replication as major factors

Unit I:

Fundamentals:

What is Distributed Computing Systems? Evolution of Distributed Computing System; Distributed Computing System Models; What is Distributed Operating System? Issues in Designing a Distributed Operating System; Introduction to Distributed Computing Environment(DCE).

Message Passing:

Introduction, Desirable features of a Good Message Passing System, Issues in PC by Message Passing, Synchronization, Buffering, Multi-datagram Messages, Encoding and Decoding of Message Data, Process Addressing, Failure Handling, Group Communication, Case Study: 4.3 BSD UNIX IPC Mechanism.

Unit II: Remote Procedure Calls:

Introduction, The RPC Model, Transparency of RPC, Implementing RPC Mechanism, Stub Generation, RPC Messages, Marshaling Arguments and Results, Server Management, Parameter-Passing Semantics, Call Semantics, Communication Protocols for RPCs, Complicated RPCs, Client-Server Binding, Exception Handling, Security, Some Special Types of RPCs, RPC in Heterogeneous Environments, Lightweight RPC, Optimization for Better Performance, Case Studies: Sun RPC

Unit III: Distributed Shared Memory:

Introduction, General Architecture of DSM systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other approaches to DSM, Heterogeneous DSM, Advantages of DSM. Synchronization: Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, Election Algorithms

Unit IV: Resource Management:

Introduction, Desirable Features of a Good Global Scheduling Algorithm, Task Assignment Approach, Load – Balancing Approach, Load – Sharing Approach Process Management: Introduction, Process Migration, Threads.

Unit V: Distributed File Systems:

Introduction, Desirable Features of a Good Distributed File System, File models, File–Accessing Models, File – Sharing Semantics, File – Caching Schemes, File Replication, Fault Tolerance, Atomic Transactions and Design Principles.

Text books

1. Pradeep. K. Sinha: Distributed Operating Systems: Concepts and Design, PHI, 2007.

Reference Books:

1. Andrew S. Tanenbaum: Distributed Operating Systems, Pearson Education, 2013.
2. Ajay D. Kshemkalyani and MukeshSinghal, Distributed Computing: Principles, Algorithms and Systems, Cambridge University Press, 2008
3. SunitaMahajan, Seema Shan, " Distributed Computing", Oxford University Press,2015

III Year II Semester	CLOUD COMPUTING LAB	L	T	P	C
		0	0	3	1.5

Course Objectives:
- To introduce the various levels of services offered by cloud.
- To give practical knowledge about working with virtualization and containers.
- To introduce the advanced concepts such as serverless computing and cloud simulation.

Course Outcomes: At the end of the course, the student should be able to
- Demonstrate various service types, delivery models and technologies of a cloud computing environment.
- Distinguish the services based on virtual machines and containers in the cloud offerings.
- Assess the challenges associated with a cloud-based application.
- Discuss advanced cloud concepts such as serverless computing and cloud simulation.
- Examine various programming paradigms suitable to solve real world and scientific problems using cloud services.

List of Experiments:
1. Lab on web services
2. Lab on IPC, messaging, publish/subscribe
3. Install Virtual Box/VMware Workstation with different flavours of Linux or windows OS on top of windows8 or above.
4. Install a C compiler in the virtual machine created using Virtual Box and execute Simple Programs.
5. Create an Amazon EC2 instance and set up a web-server on the instance and associate an IP address with the instance. In the process, create a security group allowing access to port 80 on the instance.

<div align="center">OR</div>

6. Do the same with OpenStack
7. Install Google App Engine. Create a hello world app and other simple web applications using python/java.
8. Start a Docker container and set up a web-server (e.g. apache2 or Python based Flask micro web framework) on the instance. Map the host directory as a data volume for the container.
9. Find a procedure to transfer the files from one virtual machine to another virtual machine. Similarly, from one container to another container.
10. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
11. Install Hadoop single node cluster and run simple applications like word count.

12. Utilize OpenFaaS – Serverless computing framework and demonstrate basic event driven function invocation.

13. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

Text Books:

1. Mastering Cloud Computing, 2nd edition, Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi, Shivananda Poojara, Satish N. Srirama, McGraw Hill, 2024.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.

Reference Books:

1. Cloud Computing, Theory and Practice, Dan C Marinescu, 2nd edition, MK Elsevier, 2018.
2. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
3. Online documentation and tutorials from cloud service providers (e.g. AWS, Google App Engine)
4. Docker, Reference documentation, https://docs.docker.com/reference/
5. OpenFaaS, Serverless Functions Made Simple, https://docs.openfaas.com/

III Year II Semester	CRYPTOGRAPHY & NETWORK SECURITY LAB	L	T	P	C
		0	0	3	1.5

Course Objectives:

- To learn basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.
- To understand and implement encryption and decryption using Ceaser Cipher, Substitution Cipher, Hill Cipher.

List of Experiments:

1. Write a C program that contains a string (char pointer) with a value \Hello World'. The program should XOR each character in this string with 0 and displays the result.
2. Write a C program that contains a string (char pointer) with a value \Hello World'. The program should AND or and XOR each character in this string with 127 and display the result
3. Write a Java program to perform encryption and decryption using the following algorithms:
   - a) Ceaser Cipher
   - b) Substitution Cipher
   - c) Hill Cipher
4. Write a Java program to implement the DES algorithm logic
5. Write a C/JAVA program to implement the BlowFish algorithm logic
6. Write a C/JAVA program to implement the Rijndael algorithm logic.
7. Using Java Cryptography, encrypt the text "Hello world" using BlowFish. Create your own key using Java key tool.
8. Write a Java program to implement RSA Algorithm
9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).
10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

III Year II Semester	SOFT SKILLS	L	T	P	C
		0	1	2	2

Course Objectives:
- To equip the students with the skills to effectively communicate in English
- To train the students in interview skills, group discussions and presentation skills
- To motivate the students to develop confidence
- To enhance the students' interpersonal skills
- To improve the students' writing skills

UNIT – I

Analytical Thinking & Listening Skills: Self-Introduction, Shaping Young Minds - A Talk by Azim Premji (Listening Activity), Self – Analysis, Developing Positive Attitude, Perception.

Communication Skills: Verbal Communication; Non Verbal Communication (Body Language)

UNIT – II

Self-Management Skills: Anger Management, Stress Management, Time Management, Six Thinking Hats, Team Building, Leadership Qualities

Etiquette: Social Etiquette, Business Etiquette, Telephone Etiquette, Dining Etiquette

UNIT – III

Standard Operation Methods : Basic Grammars, Tenses, Prepositions, Pronunciation, Letter Writing; Note Making, Note Taking, Minutes Preparation, Email & Letter Writing

UNIT-IV

Job-Oriented Skills: Group Discussion, Mock Group Discussions, Resume Preparation, Interview Skills, Mock Interviews

UNIT-V

Interpersonal relationships: Introduction, Importance, Types, Uses, Factors affecting interpersonal relationships, Accommodating different styles, Consequences of interpersonal relationships

Text books:
1. Barun K. Mitra, Personality Development and Soft Skills, Oxford University Press, 2011.
2. S.P. Dhanavel, English and Soft Skills, Orient Blackswan, 2010.

Reference books:

1. R.S.Aggarwal, A Modern Approach to Verbal & Non-Verbal Reasoning, S.Chand& Company Ltd., 2018.
2. Raman, Meenakshi& Sharma, Sangeeta, Technical Communication Principles and Practice, Oxford University Press, 2011.

E-resources:

1. https://swayam-plus.swayam2.ac.in/courses/course-details?id=P_CAMBR_01

III Year II Semester	**TECHNICAL PAPER WRITING & IPR**	L	T	P	C
		2	0	0	-

Course Objective : The course will explain the basic related to writing the technical reports and understanding the concepts related to formatting and structuring the report. This will help students to comprehend the concept of proofreading, proposals and practice

Unit I:

Introduction: An introduction to writing technical reports, technical sentences formation, using transitions to join sentences, Using tenses for technical writing.

Planning and Structuring: Planning the report, identifying reader(s), Voice, Formatting and structuring the report, Sections of a technical report, Minutes of meeting writing.

Unit II:

Drafting report and design issues: The use of drafts, Illustrations and graphics.

Final edits: Grammar, spelling, readability and writing in plain English: Writing in plain English, Jargon and final layout issues, Spelling, punctuation and Grammar, Padding, Paragraphs, Ambiguity.

Unit III:

Proofreading and summaries: Proofreading, summaries, Activities on summaries.

Presenting final reports: Printed presentation, Verbal presentation skills, Introduction to proposals and practice.

Unit IV: Using word processor:

Adding a Table of Contents, Updating the Table of Contents, Deleting the Table of Contents, Adding an Index, Creating an Outline, Adding Comments, Tracking Changes, Viewing Changes, Additions, and Comments, Accepting and Rejecting Changes , Working with Footnotes and Endnotes, Inserting citations and Bibliography, Comparing Documents, Combining Documents, Mark documents final and make them read only., Password protect Microsoft Word documents., Using Macros,

Unit V:

Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of **Patenting and Development:** technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property

Text Books:

1. Kompal Bansal & Parshit Bansal, "Fundamentals of IPR for Beginner's", 1st Ed., BS Publications, 2016.
2. William S. Pfeiffer and Kaye A. Adkins, "Technical Communication: A Practical Approach", Pearson.
3. Ramappa,T., "Intellectual Property Rights Under WTO", 2nd Ed., S Chand, 2015.

Reference Books:

1. Adrian Wallwork , English for Writing Research Papers, Springer New York Dordrecht Heidelberg London, 2011.
2. Day R, How to Write and Publish a Scientific Paper, Cambridge University Press(2006)

E-resources:

1. https://www.udemy.com/course/reportwriting/
2. https://www.udemy.com/course/professional-business-english-and-technical-report-writing/
3. https://www.udemy.com/course/betterbusinesswriting/

Syllabus

for

Open Electives & Minors

	PRINCIPLES OF OPERATING SYSTEMS	L	T	P	C
		3	0	0	3

Course Objectives:
The main objectives of the course is to make student
- Understand the basic concepts and principles of operating systems, including process management, memory management, file systems, and Protection
- Make use of process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
- Illustrate different conditions for deadlock and their possible solutions.

UNIT - I
Operating Systems Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Free and Open-Source Operating Systems
System Structures: Operating System Services, User and Operating-System Interface, system calls, Types of System Calls, system programs, Operating system Design and Implementation, Operating system structure, Building and Booting an Operating System, Operating system debugging

UNIT - II
Processes: Process Concept, Process scheduling, Operations on processes, Inter-process communication.
Threads and Concurrency: Multithreading models, Thread libraries, Threading issues.
CPU Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling.

UNIT – III
Synchronization Tools: The Critical Section Problem, Peterson's Solution, Mutex Locks, Semaphores, Monitors, Classic problems of Synchronization.
Deadlocks: system Model, Deadlock characterization, Methods for handling Deadlocks.

UNIT - IV
Memory-Management Strategies: Introduction, Contiguous memory allocation, Paging, Structure of the Page Table, Swapping.
Virtual Memory Management: Introduction, Demand paging, Copy-on-write, Page replacement, Allocation of frames, Thrashing
Storage Management: Overview of Mass Storage Structure, HDD Scheduling.

UNIT - V
File System: File System Interface: File concept, Access methods, Directory Structure; File system Implementation: File-system structure, File-system Operations, Directory implementation, Allocation method, Free space management; File-System Internals: File-System Mounting, Partitions and Mounting, File Sharing.

Text Books:
1. Operating System Concepts, Silberschatz A, Galvin P B, Gagne G, 10th Edition, Wiley, 2018.
2. Modern Operating Systems, Tanenbaum A S, 4th Edition, Pearson , 2016

Reference Books:
1. Operating Systems: A Concept Based Approach, D.M Dhamdhere, 3rd Edition, McGraw- Hill, 2013

Online Learning Resources:
1. https://nptel.ac.in/courses/106/106/106106144/
2. http://peterindia.net/OperatingSystems.html

	COMPUTER ORGANIZATION AND ARCHITECTURE	L	T	P	C
		3	0	0	3

Course Objectives:

The purpose of the course is to introduce principles of computer organization and the basic architectural concepts. It provides an in depth understanding of basic organization, design, programming of a simple digital computer, computer arithmetic, instruction set design, micro programmed control unit, pipelining and vector processing, memory organization and I/O systems

UNIT I:

Basic Structure Of Computers: Computer Types, Functional unit, Basic Operational concepts, Bus structures, Software, Performance, multiprocessors and multi computers.

Logic gates: Digital Logic gates,Two-level realizations using gates - AND-OR, OR-AND, NAND-NAND and NOR-NOR

UNIT II:

Sequential circuits I: Classification of sequential circuits (synchronous and asynchronous): basic flip-flops, truth tables and excitation tables (NAND RS latch, NOR RS latch, RS flip-flop. JK flip-flop, T flip-flop, D flip-flop with reset and clear terminals).Conversion of flip-flop to flip-flop, Race around condition, Master J-K flipflop

Register Transfer Language And Micro-operations: Register Transfer language. Register Transfer Bus and memory transfers, Arithmetic Micro-operations, Logic micro operations, shift micro operations, Arithmetic logic shift unit. Instruction codes. Computer Registers, Computer instructions, Instruction cycle.

UNIT III:

Micro Programmed Control: Control memory, Address sequencing, micro program example, design of control unit.

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

UNIT IV:

Microprocessors: Evaluation of Microprocessors, CISC and RISC, Characteristics of Microprocessors

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory, Cache memories performance considerations, Virtual memories Introduction to Shift registers and RAID

UNIT V:

Input – Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupts, DMA, Input Output Processor, Serial Communication.

Text Books:

1. Digital Logic and Computer Design, Moriss Mano, 11th Edition, Pearson Education.
2. Computer Organization, 5thed.,Hamacher, Vranesicand Zaky,TMH,2002
3. Computer System Architecture, 3/e, MorisMano, Pearson/PHI.

Reference Books:

1. Computer System Organization &Architecture, John D.Carpinelli, Pearson,2008
2. Computer System Organization, NareshJotwani, TMH, 2009
3. Computer Organization & Architecture: Designing for Performance, 7th ed., William Stallings, PHI, 2006
4. Structured Computer Organization, Andrew S. Tanenbaum, 4th Edition, PHI/Pearson.

	PRINCIPLES OF DATABASE MANAGEMENT SYSTEMS	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of the course is to

- Introduce database management systems and to give a good formal foundation on the relational model of data
- Introduce the concepts of SQL
- Demonstrate the principles behind systematic database design approaches by covering conceptual design, logical design through normalization
- Provide an overview of physical design of a database system, by discussing database storage techniques

UNIT I:

Introduction: Database system, Characteristics (Database Vs File System), Database Users, Advantages of Database systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

UNIT II:

Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams.

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance.

UNIT III:

BASIC SQL: Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update).

SQL: Basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions(Date and Time, Numeric, String conversion).Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view(updatable and non-updatable), relational set operations.

UNIT IV:

Schema Refinement (Normalization): Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency Lossless join and dependency preserving decomposition, (1NF, 2NF and 3 NF), concept of surrogate key, Boyce-Codd normal form (BCNF).

UNIT V:

Transaction Concept: Transaction State, ACID properties, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, lock based, time stamp based, optimistic, concurrency protocols, Deadlocks, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm.

Text Books:
1) Database Management Systems, 3rd edition, Raghurama Krishnan, Johannes Gehrke, TMH (For Chapters 2, 3, 4)
2) Database System Concepts,5th edition, Silberschatz, Korth, Sudarsan, TMH (For Chapter 1 and Chapter 5)

Reference Books:
1) Introduction to Database Systems, 8thedition, C J Date, Pearson.
2) Database Management System, 6th edition, Ramez Elmasri, Shamkant B. Navathe, Pearson
3) Database Principles Fundamentals of Design Implementation and Management, 10th edition, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning, 2022

Web-Resources:
1) https://nptel.ac.in/courses/106/105/106105175/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012758066672820 22456_shared/overview

	PRINCIPLES OF DATABASE	**L**	**T**	**P**	**C**
	MANAGEMENT SYSTEMS LAB	**0**	**0**	**3**	**1.5**

Course Objectives:

This Course will enable students to
- Populate and query a database using SQL DDL/DML Commands
- Declare and enforce integrity constraints on a database
- Writing Queries using advanced concepts of SQL
- Programming PL/SQL including procedures, functions, cursors and triggers

Experiments covering the topics:
- DDL, DML, DCL commands
- Queries, nested queries, built-in functions,
- PL/SQL programming- control structures
- Procedures, Functions, Cursors, Triggers,

Sample Experiments:
1. Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables), examples using SELECT command.
2. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.
   Example:- Select the roll number and name of the student who secured fourth rank in the class.
3. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
4. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)
5.

   i. Create a simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
   ii. Insert data into student table and use COMMIT, ROLLBACK and
   SAVEPOINT in PL/SQL block.
6. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
7. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE-APPLICATION ERROR.
8. Program development using creation of procedures, passing parameters IN and OUT of PROCEDURES.

9. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
10. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
11. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

Text Books/Suggested Reading:
1. Oracle: The Complete Reference by Oracle Press
2. Nilesh Shah, "Database Systems Using Oracle", PHI, 2007
3. Rick F Vander Lans, "Introduction to SQL", Fourth Edition, Pearson Education, 2007

OBJECT ORIENTED PROGRAMMING THROUGH JAVA	L	T	P	C
	3	0	0	3

Course Objectives:

The learning objectives of this course are to:

- identify Java language components and how they work together in applications
- Learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries.
- learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications
- understand how to design applications with threads in Java
- understand how to use Java APIs for program development

UNIT I:

Object Oriented Programming: Basic concepts, Principles,

Program Structure in Java: Introduction, Writing Simple Java Programs, Elements or Tokens in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences Comments, Programming Style.

Data Types, Variables, and Operators :Introduction, Data Types in Java, Declaration of Variables, Data Types, Type Casting, Scope of Variable Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf() Method, Static Variables and Methods, Attribute Final, Introduction to Operators, Precedence and Associativity of Operators, Assignment Operator (=), Basic Arithmetic Operators, Increment (++) and Decrement (--) Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators.

Control Statements: Introduction, if Expression, Nested if Expressions, if–else Expressions, Ternary Operator?:, Switch Statement, Iteration Statements, while Expression, do–while Loop, for Loop, Nested for Loop, For–Each for Loop, Break Statement, Continue Statement.

UNIT II:

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing Private Members of Class, Constructor Methods for Class, Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by Reference, Keyword this.

Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static

UNIT III:

Arrays: Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Sorting of Arrays, Search for Values in Arrays, Class Arrays, Two-dimensional Arrays, Arrays of Varying Lengths, Three dimensional Arrays, Arrays as Vectors.

Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class-Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Inheritance.

Interfaces: Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface, Functional Interfaces, Annotations.

UNIT IV:
Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in Java SE, Java.lang Package and its Classes, Class Object, Enumeration, class Math, Wrapper Classes, Autoboxing and Auto-unboxing, Java util Classes and Interfaces, Formatter Class, Random Class, Time Package, Class Instant (java.time.Instant), Formatting for Date/Time in Java, Temporal Adjusters Class, Temporal Adjusters Class.
Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions.
Java I/O and File: Java I/O API, standard I/O streams, types, Byte streams, Character streams, Scanner class, Files in Java(Text Book 2)

UNIT V:
String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings,Comparison, Modifying, Searching; Class String Buffer.
Multithreaded Programming: Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread-Creation of New Threads, Thread States, Thread Priority-Synchronization, Deadlock and Race Situations, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads.
Java Database Connectivity: Introduction, JDBC Architecture, Installing MySQL and MySQL Connector/J, JDBC Environment Setup, Establishing JDBC Database Connections, Result Set Interface
Java FX GUI: Java FX Scene Builder, Java FX App Window Structure, displaying text and image, event handling, laying out nodes in scene graph, mouse events (Text Book 3)

Text Books:
1) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford.
2) Joy with JAVA, Fundamentals of Object Oriented Programming, Debasis Samanta, Monalisa Sarma, Cambridge, 2023.
3) JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

References Books:
1) The complete Reference Java, 11thedition, Herbert Schildt,TMH
2) Introduction to Java programming, 7th Edition, Y Daniel Liang, Pearson

Online Resources:
1) https://nptel.ac.in/courses/106/105/106105191/
2)
https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview

ADVANCED DATA STRUCTURES &	L	T	P	C
ALGORITHM ANALYSIS	3	0	0	3

Course Objectives:
The main objectives of the course is to
- provide knowledge on advance data structures frequently used in Computer Science domain
- Develop skills in algorithm design techniques popularly used
- Understand the use of various data structures in the algorithm design

UNIT – I:
Introduction to Algorithm Analysis, Space and Time Complexity analysis, Asymptotic Notations.
AVL Trees – Creation, Insertion, Deletion operations and Applications
B-Trees – Creation, Insertion, Deletion operations and Applications

UNIT – II:
Heap Trees (Priority Queues) – Min and Max Heaps, Operations and Applications
Graphs – Terminology, Representations, Basic Search and Traversals, Connected Components and Biconnected Components, applications
Divide and Conquer: The General Method, Quick Sort, Merge Sort, Strassen's matrix multiplication, Convex Hull

UNIT – III:
Greedy Method: General Method, Job Sequencing with deadlines, Knapsack Problem, Minimum cost spanning trees, Single Source Shortest Paths
Dynamic Programming: General Method, All pairs shortest paths, Single Source Shortest Paths– General Weights (Bellman Ford Algorithm), Optimal Binary Search Trees, 0/1 Knapsack, String Editing, Travelling Salesperson problem

UNIT – IV:
Backtracking: General Method, 8-Queens Problem, Sum of Subsets problem, Graph Coloring, 0/1 Knapsack Problem
Branch and Bound: The General Method, 0/1 Knapsack Problem, Travelling Salesperson Problem

UNIT – V:
NP Hard and NP Complete Problems: Basic Concepts, Cook's theorem
NP Hard Graph Problems: Clique Decision Problem (CDP), Chromatic Number Decision Problem (CNDP), Traveling Salesperson Decision Problem (TSP)
NP Hard Scheduling Problems: Scheduling Identical Processors, Job Shop Scheduling

Textbooks:
1. Fundamentals of Data Structures in C++, Horowitz, Ellis; Sahni, Sartaj; Mehta, Dinesh, 2ndEdition Universities Press
2. Computer Algorithms in C++, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2nd Edition University Press

Reference Books:
1. Data Structures and program design in C, Robert Kruse, Pearson Education Asia
2. An introduction to Data Structures with applications, Trembley& Sorenson, McGraw Hill
3. The Art of Computer Programming, Vol.1: Fundamental Algorithms, Donald E Knuth, Addison-Wesley, 1997.
4. Data Structures using C & C++: Langsam, Augenstein&Tanenbaum, Pearson, 1995
5. Algorithms + Data Structures & Programs:, N.Wirth, PHI
6. Fundamentals of Data Structures in C++: Horowitz Sahni& Mehta, Galgottia Pub.
7. Data structures in Java:, Thomas Standish, Pearson Education Asia

Online Learning Resources:
1. https://www.tutorialspoint.com/advanced_data_structures/index.asp
2. http://peterindia.net/Algorithms.html
3. Abdul Bari, Introduction to Algorithms (youtube.com)

	ADVANCED DATA STRUCTURES & ALGORITHM ANALYSIS LAB	**L**	**T**	**P**	**C**
		0	0	3	1.5

Course Objectives:

The objectives of the course is to

- acquire practical skills in constructing and managing Data structures
- apply the popular algorithm design methods in problem-solving scenarios

Experiments covering the Topics:

- Operations on AVL trees, B-Trees, Heap Trees
- Graph Traversals
- Sorting techniques
- Minimum cost spanning trees
- Shortest path algorithms
- 0/1 Knapsack Problem
- Travelling Salesperson problem
- Optimal Binary Search Trees
- N-Queens Problem
- Job Sequencing

Sample Programs:

1. Construct an AVL tree for a given set of elements which are stored in a file. And implement insert and delete operation on the constructed tree. Write contents of tree into a new file using in-order.
2. Construct B-Tree an order of 5 with a set of 100 random elements stored in array. Implement searching, insertion and deletion operations.
3. Construct Min and Max Heap using arrays, delete any element and display the content of the Heap.
4. Implement BFT and DFT for given graph, when graph is represented by
   a) Adjacency Matrix b) Adjacency Lists
5. Write a program for finding the biconnected components in a given graph.
6. Implement Quick sort and Merge sort and observe the execution time for various input sizes (Average, Worst and Best cases).
7. Compare the performance of Single Source Shortest Paths using Greedy method when the graph is represented by adjacency matrix and adjacency lists.
8. Implement Job Sequencing with deadlines using Greedy strategy.
9. Write a program to solve 0/1 Knapsack problem Using Dynamic Programming.
10. Implement N-Queens Problem Using Backtracking.
11. Use Backtracking strategy to solve 0/1 Knapsack problem.
12. Implement Travelling Sales Person problem using Branch and Bound approach.

Reference Books:
1. Fundamentals of Data Structures in C++, Horowitz Ellis, Sahni Sartaj, Mehta, Dinesh, 2nd Edition, Universities Press
2. Computer Algorithms/C++ Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2nd Edition, University Press
3. Data Structures and program design in C, Robert Kruse, Pearson Education Asia
4. An introduction to Data Structures with applications, Trembley & Sorenson, McGraw Hill

Online Learning Resources:
1. http://cse01-iiith.vlabs.ac.in/
2. http://peterindia.net/Algorithms.html

PRINCIPLES OF SOFTWARE ENGINEERING			L	T	P	C
			3	0	0	3

Course Objectives:

The objectives of this course are to introduce
- Software life cycle models, Software requirements and SRS document.
- Project Planning, quality control and ensuring good quality software.
- Software Testing strategies, use of CASE tools, Implementation issues, validation & verification procedures.

UNIT I:

Introduction: Evolution, Software development projects, Exploratory style of software developments, Emergence of software engineering, Notable changes in software development practices, Computer system engineering.

Software Life Cycle Models: Basic concepts, Waterfall model and its extensions, Rapid application development, Agile development model, Spiral model.

UNIT II:

Software Project Management: Software project management complexities, Responsibilities of a software project manager, Metrics for project size estimation, Project estimation techniques, Empirical Estimation techniques, COCOMO, risk management.

Requirements Analysis and Specification: Requirements gathering and analysis, Software Requirements Specification (SRS), Formal system specification, Axiomatic specification, Algebraic specification, Executable specification and 4GL.

UNIT III:

Software Design: Overview of the design process, How to characterize a good software design? Layered arrangement of modules, Cohesion and Coupling. approaches to software design.

Agility: Agility and the Cost of Change, Agile Process, Extreme Programming (XP), Other Agile Process Models, Tool Set for the Agile Process (Text Book 2)

Function-Oriented Software Design: Overview of SA/SD methodology, Structured analysis, Developing the DFD model of a system, Structured design, Detailed design, and Design Review.

User Interface Design: Characteristics of a good user interface, Basic concepts, Types of user interfaces, Fundamentals of component-based GUI development, and user interface design methodology.

UNIT IV:

Coding and Testing: Coding, Code review, Software documentation, Testing, Black-box testing, White-Box testing, Debugging, Program analysis tools, Integration testing, Testing object-oriented programs, Smoke testing, and Some general issues associated with testing.

Software Reliability and Quality Management: Software reliability. Statistical testing, Software quality, Software quality management system, ISO 9000. SEI Capability maturity model. Few other important quality standards, and Six Sigma**.**

UNIT V:
Software Maintenance: Characteristics of software maintenance, Software reverse engineering, Software maintenance process models and Estimation of maintenance cost.
Software Reuse: reuse- definition, introduction, reason behind no reuse so far, Basic issues in any reuse program, A reuse approach, and Reuse at organization level.

Text Books:
1. Fundamentals of Software Engineering, Rajib Mall, 5th Edition, PHI.
2. Software Engineering: A Practitioner's Approach, Roger S. Pressman, 9th Edition, Mc-Graw Hill International Edition.

Reference Books:
1. Software Engineering, Ian Sommerville,10th Edition, Pearson.
2. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.

e-Resources:
1) https://nptel.ac.in/courses/106/105/106105182/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012605895063871 48827_shared/overview
3) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013382690411003 904735_shared/overview

	COMPUTER NETWORKS	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Objectives:
- To provide insight about networks, topologies, and the key concepts.
- To gain comprehensive knowledge about the layered communication architectures (OSI and TCP/IP) and its functionalities.
- To understand the principles, key protocols, design issues, and significance of each layers in ISO and TCP/IP.
- To know the basic concepts of network services and various network applications.

UNIT I: Introduction: Network Types, LAN, MAN, WAN, Network Topologies Reference models- The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models, OSI Vs TCP/IP.
Physical Layer –Introduction to Guided Media- Twisted-pair cable, Coaxial cable and Fiber optic cable and introduction about unguided media.

UNIT II: Data link layer: Design issues, **Framing**: fixed size framing, variable size framing, flow control, error control, error detection and correction codes, CRC, Checksum: idea, one's complement internet checksum, services provided to Network Layer, **Elementary Data Link Layer protocols**: simplex protocol, Simplex stop and wait, Simplex protocol for Noisy Channel.
Sliding window protocol: One bit, Go back N, Selective repeat-Stop and wait protocol, Data link layer in HDLC, Point to point protocol (PPP)

UNIT – III: Media Access Control: Random Access: ALOHA, Carrier sense multiple access (CSMA), CSMA with Collision Detection, CSMA with Collision Avoidance, **Controlled Access:** Reservation, Polling, Token Passing, **Channelization:** frequency division multiple Access(FDMA), time division multiple access(TDMA), code division multiple access(CDMA).
Wired LANs: Ethernet, Ethernet Protocol, Standard Ethernet, Fast Ethernet(100 Mbps), Gigabit Ethernet, 10 Gigabit Ethernet.

UNIT – IV: The Network Layer Design Issues – Store and Forward Packet Switching-Services Provided to the Transport layer- Implementation of Connectionless Service-Implementation of Connection Oriented Service- Comparison of Virtual Circuit and Datagram Networks,
Routing Algorithms-The Optimality principle-Shortest path, Flooding, Distance vector, Link state, Hierarchical, Congestion Control algorithms-General principles of congestion control, Congestion prevention polices, Approaches to Congestion Control-Traffic Aware Routing-Admission Control-Traffic Throttling-Load Shedding. Traffic Control Algorithm-Leaky bucket & Token bucket.

Internet Working: How networks differ- How networks can be connected- Tunnelling, internetwork routing-, Fragmentation, network layer in the internet – IP protocols-IP Version 4 protocol-IPV4 Header Format, IP addresses, Class full Addressing, CIDR, Subnets-IP Version 6-The main IPV6 header, Transition from IPV4 to IPV6, Comparison of IPV4 & IPV6.

UNIT –V: The Transport Layer: Transport layer protocols: Introduction-services- port number-User data gram protocol-User datagram-UDP services-UDP applications-Transmission control protocol: TCP services- TCP features- Segment- A TCP connection-windows in TCP- flow control-Error control, Congestion control in TCP.

Application Layer –- World Wide Web: HTTP, Electronic mail-Architecture- web based mail- email security- TELENET-local versus remote Logging-Domain Name System.

Text Books:
3. Computer Networksm, Andrew S Tanenbaum, Fifth Edition. Pearson Education/PHI
4. Data Communications and Networks, Behrouz A. Forouzan, Fifth Edition TMH.

References Books:
3. Data Communications and Networks- Achut S Godbole, AtulKahate
4. Computer Networks, Mayank Dave, CENGAGE

B.Tech. IV Year I Semester

S. No.	Category	Title	L	T	P	C
1	Professional Core	Deep Learning	2	1	0	3
2	Management Course- II	Human Resources & Project Management	2	0	0	2
3	Professional Elective-IV	1. Software Architecture & Design Patterns 2. Blockchain Technology 3. Augmented Reality & Virtual Reality 4. Internet of Things 5. Agentic AI 6. 12 week MOOC Swayam/NPTEL course recommended by the BoS	3	0	0	3
4	Professional Elective-V	1. Agile Methodologies 2. Generative AI 3. Computer Vision 4. Cyber Physical Systems 5. 12 week MOOC Swayam / NPTEL course recommended by the BoS	3	0	0	3
5	Open Elective-III		3	0	0	3
6	Open Elective-IV		3	0	0	3
7	Skill Enhancement Course	Prompt Engineering/ SWAYAM Plus - Certificate program in Prompt Engineering and ChatGPT	0	1	2	2
8	Audit Course	Constitution of India	2	0	0	-
9	Internship	Evaluation of Industry Internship /Mini Project	-	-	-	2
Total			**18**	**2**	**02**	**21**
MC	Minor Course (Student may select from the same specialized minors pool)		3	0	0	3
HC	Honors Course (Student may select from the same honors pool)		3	0	0	3
HC	Honors Course (Student may select from the same honors pool)		3	0	0	3

B.Tech. IV Year II Semester

S. No.	Category	Title	L	T	P	C
1	Internship & Project Work	Full semester Internship & Project Work	0	0	24	12

Note: *Student need to do at least ONE MOOC/NPTEL Course (of 3 credits out of 160 credits) to meet the mandatory requirement (11th criteria, as per R23 Regulations); they are allowed to register one semester in advance*

Open Electives, offered to other department students:

Open Elective I: Principles of Operating Systems/ Computer Organization and
                  Architecture

Open Elective II: Principles of Database Management Systems

Open Elective III: Object Oriented Programming Through Java

Open Elective IV: Principles of Software Engineering /Computer Networks

Minor Engineering

Note:
1. *To obtain Minor Engineering, student needs to obtain 18 credits by successfully completing any of the following courses in the concern stream.*
2. *During Minor/Honors Course selection, there should not be any overlapping with Regular/Major/OPEN Electives*

Minor in CSE
1. Principles of Database Management Systems   3-0-3-4.5 (II-II)
2. Principles of Software Engineering   3-0-0-3 (III-I)
3. Advanced Data Structures & Algorithm Analysis   3-0-3-4.5 (III-II)
4. Principles of Operating Systems   3-0-0-3 (IV-I)
    Any of the following 12 Week 3 credit NPTEL MOOC Courses
5. Artificial Intelligence: Knowledge Representation and Reasoning
6. Computer Networks and Internet Protocol
7. Machine Learning and Deep Learning - Fundamentals and Applications
8. Fundamentals of Object Oriented Programming
9. Discrete Mathematics for CS

COURSES OFFERED FOR HONORS DEGREE IN CSE

Note: *To obtain Honor's degree, student needs to obtain 18 credits by successfully completing any of the following courses in the concern stream. (without duplication)*

Suggested MOOC Courses for HONORS in CSE

Student need to obtain 18 Credits by successfully completing the following
  Mandatory Course(s)
1. Parallel Computer Architecture (MOOCS- SWAYAM / NPTEL 12W)
2. Quantum Algorithms and Cryptography (MOOCS- SWAYAM / NPTEL 12W)

  Any of the following for remaining 12 Credits
3. Deep Learning for Computer Vision (MOOCS- SWAYAM / NPTEL 12W)
4. Applied Linear Algebra in AI & ML (MOOCS- SWAYAM / NPTEL 12W)
5. Applied Time-Series Analysis (MOOCS- SWAYAM / NPTEL 12W)
6. Machine Learning for Engineering and Science Applications (MOOCS- SWAYAM / NPTEL 12W)
7. Practical High-Performance Computing (MOOCS- SWAYAM / NPTEL 12W)
8. Deep Learning for Natural Language Processing (MOOCS- SWAYAM / NPTEL 12W)
9. Privacy and Security in Online Social Media (MOOCS- SWAYAM / NPTEL 12W)
10. Natural Language Processing (MOOCS- SWAYAM / NPTEL 12W)

IV Year I Semester	DEEP LEARNING	L	T	P	C
		2	1	0	3

Course Objectives:

The main objective of the course is to make students:

- Learn deep learning methods for working with sequential data, deep recurrent & memory networks and deep Turing machines,
- Apply such deep learning mechanisms to various learning problems.
- Identify the open issues in deep learning, and have a grasp of the current research directions.

UNIT I:

Fundamentals of Deep Learning: Artificial Intelligence, History of Machine learning: Probabilistic Modeling, Early Neural Networks, Kernel Methods, Decision Trees, Random forests and Gradient Boosting Machines, **Fundamentals of Machine Learning:** Four Branches of Machine Learning, Evaluating Machine learning Models, Overfitting and Underfitting. **[Text Book 2]**

UNIT II:

Introducing Deep Learning: Biological and Machine Vision, Human and Machine Language, Artificial Neural Networks, Training Deep Networks, Improving Deep Networks. **[Text Book3]**

UNIT III:

Neural Networks: Anatomy of Neural Network, Introduction to Keras: Keras, TensorFlow, Theano and CNTK, Setting up Deep Learning Workstation, Classifying Movie Reviews: Binary Classification, Classifying newswires: Multiclass Classification. **[Text Book 2]**

UNIT IV:

Convolutional Neural Networks: Nerual Network and Representation Learing, Convolutional Layers, Multichannel Convolution Operation, **Recurrent Neural Networks:** Introduction to RNN, RNN Code, PyTorch Tensors: Deep Learning with PyTorch, CNN in PyTorch. **[Text Book 3]**

UNIT V:

Interactive Applications of Deep Learning: Machine Vision, Natural Language processing, Generative Adversial Networks, Deep Reinforcement Learning. **[Text Book 1]**

Deep Learning Research: Autoencoders**,** Deep Generative Models: Boltzmann Machines Restricted Boltzmann Machines, Deep Belief Networks. **[Text Book 1]**

Text Books:

1. Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courvile, MIT Press, 2016
2. Deep Learning with Python, Francois Chollet, December 2017, Manning Publications, ISBN: 9781617294433

3. Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence, Jon Krohn, Grant Beyleveld, Aglaé Bassens, September 2019, Addison-Wesley Professional, ISBN: 9780135116821

4. Deep Learning from Scratch, Seth Weidman, September 2019, O'Reilly Media, Inc., ISBN: 9781492041412

Reference Books:

1. Artificial Neural Networks, Yegnanarayana, B., PHI Learning Pvt. Ltd, 2009.
2. Matrix Computations, Golub, G.,H., Van Loan,C.,F, JHU Press,2013.
3. Neural Networks: A Classroom Approach, Satish Kumar, Tata McGraw-Hill Education, 2004.

Web Link:

1. Swayam NPTEL: Deep Learning:
   https://onlinecourses.nptel.ac.in/noc22_cs22/preview

IV Year I Semester	HUMAN RESOURCES & PROJECT MANAGEMENT	L	T	P	C
		2	0	0	2

Course Objectives:
The main objectives of the course are to
- Provide knowledge about HR planning, recruitment, selection, and job design.
- Develop skills in managing HR functions such as performance appraisal, compensation, and employee relations.
- Emphasize the importance of ethical practices and HR audits in maintaining organizational health.
- Understand the HRD framework and its impact on organizational success.
- Improve group interaction and team dynamics for better collaboration and performance.
- Understand the Fundamentals of Project Management and Project Networks
- Implement appropriate management strategies tailored to specific challenges in different project types.

UNIT –I:
HRM: Nature, Scope, Concept of HRM, Functions of HRM, Role of HR manager, emerging trends in HRM, E-HRM, HR audit models, ethical aspects of HRM. HR Planning, Demand and Supply forecasting of HR, Job Design, Recruitment, Sources of recruitment, Selection-Selection Procedure.

UNIT –II:
HRD, HR accounting, Models, Concept of Training and Development, Methods of Training. Performance Appraisal: Importance Methods of performance appraisal, Career Development and Counseling, group interaction.

UNIT –III:
Basics of Project Management, Concept, resource management, Project environment, Types of Projects, project networks-DPR, Project life cycle, Project proposals, Monitoring project progress, Project appraisal and Project selection, 80-20 rules, production technology, communication matrix

UNIT-IV:
Identify various project types and their unique management challenges and apply appropriate management strategies for each. Project Implementation and Review: Forms of project organization, project planning, project control, human aspects of project management, prerequisites for successful project implementation, project review, performance evaluation, abandonment analysis

UNIT-V:
Project Implementation and Review: Forms of project organization, project planning, project control, human aspects of project management, prerequisites for successful project implementation, project review, performance evaluation, abandonment analysis

Text Books:
1. Robert L. Mathis, John H. Jackson, Manas Ranjan Tripathy, Human Resource Management, Cengage Learning 2016.
2. Sharon Pande and Swapnalekha Basak, Human Resource Management, Text and Cases, Vikas Publishing, 2e, 2016.
3. Stewart R. Clegg, Torgeir Skyttermoen, Anne Live Vaagaasar, Project Management, Sage Publications, 1e, 2021.
4. K. Nagarajan, Project Management, New Age International Publishers, 8e, 2017.

Reference Books :
1. Subba Rao P, "Personnel and Human Resource Management-Text and Cases", Himalaya Publications, Mumbai, 2013.
2. K Aswathappa, "Human Resource and Personnel Management", Tata McGraw Hill, New Delhi, 2013.
3. Prasanna Chandra, "Projects, Planning, Analysis, Selection, Financing, Implementation and Review", Tata McGraw Hill Company Pvt. Ltd., New Delhi, 1998.
4. Vasanth Desai, "Project Management", 4th edition, Himalaya Publications, 2018.
5. Lalitha Balakrishnan, Gowri, "Project Management", Himalaya publishing house, New Delhi, 2022.

IV Year I Semester	**SOFTWARE ARCHITECTURE & DESIGN PATTERNS (PROFESSIONAL ELECTIVE-IV)**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Objectives:

The main objectives of the course are to make student

- Understand the basic concepts to identify state behavior of real world objects
- Apply Object Oriented Analysis and Design concepts to solve complex problems
- Construct various UML models using the appropriate notation for specific problem context
- Design models to Show the importance of systems analysis and design in solving complex problems using case studies
- Study Pattern Oriented approach for real world problems

UNIT – I:

Introduction: design pattern, Describing design patterns, the catalog of design pattern, organizing the catalog, how design patterns solve design problems, how to select a design pattern, how to use a design pattern What is object oriented development? key concepts of object oriented design other related concepts, benefits and drawbacks of the paradigm

UNIT – II:

Analysis a System: Overview of the analysis phase, stage 1 gathering the requirements functional requirements specification, defining conceptual classes and relationships, using the knowledge of the domain Design and Implementation, discussions and further reading

UNIT – III:

Design Pattern Catalog: Structural patterns, Adapter, bridge, composite, decorator, facade, flyweight, proxy.

UNIT – IV:

Interactive systems and the MVC architecture: Introduction The MVC architectural pattern, analyzing a simple drawing program designing the system, designing of the subsystems, getting into implementation, implementing undo operation drawing incomplete items, adding a new feature pattern based solutions

UNIT – V :

Designing with Distributed Objects: Client server system, java remote method invocation, implementing an object oriented system on the web, Web services (SOAP, Restful), Enterprise Service Bus

Text Books:

1. Object Oriented Analysis, Design and Implementation, Brahma Dathan, Sarnath Rammath , Universities Press, 2013
2. Design Patterns, Erich Gamma, Richard Helan, Ralph Johman, John Vlissides, PEARSON Publication, 2013

Reference Books:

1. Frank Bachmann, Regine Meunier, Hans Rohnert "Pattern Oriented Software Architecture", Volume 1, 1996.
2. William J Brown et al., "Anti Patterns: Refactoring Software, Architectures and Projects in Crisis", John Wiley, 1998

IV Year I Semester	BLOCKCHAIN TECHNOLOGY (PROFESSIONAL ELECTIVE-IV)	L	T	P	C
		3	0	0	3

Course Objectives:

The objectives of the course are to make student

1. Learn the fundamentals of Block Chain and various types of block chain and consensus mechanism.
2. Understand public block chain system, Private block chain system and consortium block chain.
3. Identify the security issues of blockchain technology.

UNIT – I:

Fundamentals of Blockchain: Introduction, Origin of Blockchain, Blockchain Solution, Components of Blockchain, Block in a Blockchain, The Technology and the Future.

Blockchain Types and Consensus Mechanism: Introduction, Decentralization and Distribution, Types of Blockchain, Consensus Protocol.

Cryptocurrency: Bitcoin, Altcoin and Token: Introduction, Bitcoin and the Cryptocurrency, Cryptocurrency Basics, Types of Cryptocurrencies, Cryptocurrency Usage.

UNIT – II:

Public Blockchain System: Introduction, Public Blockchain, Popular Public Blockchains, The Bitcoin Blockchain, Ethereum Blockchain.

Smart Contracts: Introduction, Smart Contract, Characteristics of a Smart Contract, Types of Smart Contracts, Types of Oracles, Smart Contracts in Ethereum, Smart Contracts in Industry.

UNIT – III:

Private Blockchain System: Introduction, Key Characteristics of Private Blockchain, Private Blockchain, Private Blockchain Examples, Private Blockchain and Open Source, E-commerce Site Example, Various Commands (Instructions) in E-commerce Blockchain, Smart Contract in Private Environment, State Machine, Different Algorithms of Permissioned Blockchain, Byzantine Fault, Multichain.

Consortium Blockchain: Introduction, Key Characteristics of Consortium Blockchain, Need of Consortium Blockchain, Hyperledger Platform, Overview of Ripple, Overview of Corda.

Initial Coin Offering: Introduction, Blockchain Fundraising Methods, Launching an ICO, Investing in an ICO, Pros and Cons of Initial Coin Offering, Successful Initial Coin Offerings, Evolution of ICO, ICO Platforms.

UNIT – IV:

Security in Blockchain: Introduction, Security Aspects in Bitcoin, Security and Privacy Challenges of Blockchain in General, Performance and Scalability, Identity Management and Authentication, Regulatory Compliance and Assurance, Safeguarding Blockchain Smart Contract (DApp), Security Aspects in Hyperledger Fabric.

Applications of Blockchain: Introduction, Blockchain in Banking and Finance, Blockchain in Education, Blockchain in Energy, Blockchain in Healthcare, Blockchain in Real-estate, Blockchain in Supply Chain, The Blockchain and IoT. Limitations and Challenges of Blockchain.

UNIT – V:
Blockchain Case Studies:
Case Study 1 – Retail,
Case Study 2 – Banking and Financial Services,
Case Study 3 – Healthcare,
Case Study 4 – Energy and Utilities.
Blockchain Platform using Python: Introduction, Learn How to Use Python Online Editor, Basic Programming Using Python, Python Packages for Blockchain.
Blockchain platform using Hyperledger Fabric: Introduction, Components of Hyperledger Fabric Network, Chain codes from Developer.ibm.com, Blockchain Application Using Fabric Java SDK.

Text book:
1. "Blockchain Technology", Chandramouli Subramanian, Asha A.George, Abhilasj K A, Meena Karthikeyan , Universities Press.

Reference Books:
1. Blockchain Blue print for Economy, Melanie Swan, SPD Oreilly.
2. Blockchain for Business, Jai Singh Arun, Jerry Cuomo, Nitin Gauar, Pearson Addition Wesley

IV Year I Semester	AUGMENTED REALITY & VIRTUAL REALITY (PROFESSIONAL ELECTIVE-IV)	L	T	P	C
		3	0	0	3

Course Objectives:

The objectives of the course are to
- Provide a foundation to the fast growing field of AR and make the students aware of the various Augmented Reality concepts.
- Give historical and modern overviews and perspectives on virtual reality.

UNIT – I:

Introduction to Augmented Reality: Augmented Reality, Defining augmented reality, history of augmented reality, Examples, Related fields

Displays: Multimodal Displays, Visual Perception, Requirements and Characteristics, Spatial Display Model, Visual Displays

Tracking: Tracking, Calibration, and Registration, Coordinate Systems, Characteristics of Tracking Technology, Stationary Tracking Systems, Mobile Sensors

UNIT – II:

Computer Vision for Augmented Reality: Marker Tracking, Multiple-Camera Infrared Tracking, Natural Feature Tracking by Detection, Outdoor Tracking.

Interaction: Output Modalities, Input Modalities, Tangible Interfaces, Virtual User Interfaces on Real Surfaces, Augmented Paper, Multi-view Interfaces, Haptic Interaction

Software Architectures: AR Application Requirements, Software Engineering Requirements, Distributed Object Systems, Dataflow, Scene Graphs

UNIT – III:

Introduction to Virtual Reality: Defining Virtual Reality, History of VR, Human Physiology and Perception

The Geometry of Virtual Worlds: Geometric Models, Axis-Angle Representations of Rotation, Viewing Transformations

Light and Optics: Basic Behavior of Light, Lenses, Optical Aberrations, The Human Eye, Cameras, Displays

UNIT – IV:

The Physiology of Human Vision: From the Cornea to Photoreceptors, From Photoreceptors to the Visual Cortex, Eye Movements, Implications for VR

Visual Perception: Visual Perception - Perception of Depth, Perception of Motion, **Perception of Color Visual Rendering:** Visual Rendering -Ray Tracing and Shading Models, Rasterization, Correcting Optical Distortions, Improving Latency and Frame Rates, Immersive Photos and Videos

UNIT – V:

Motion in Real and Virtual Worlds: Velocities and Accelerations, The Vestibular System, Physics in the Virtual World, Mismatched Motion and Vection

Interaction: Motor Programs and Remapping, Locomotion, Social Interaction

Audio: The Physics of Sound, The Physiology of Human Hearing, Auditory Perception, Auditory Rendering

Text Books:

1. "Augmented Reality: Principles & Practice" by Schmalstieg, Hollerer, Pearson Education India; First edition (12 October 2016), ISBN-10: 9332578494
2. Virtual Reality, Steven M. LaValle, Cambridge University Press, 2016

Reference Books:
1. "AR Game Development", Allan Fowler, 1st Edition, Apress Publications, 2018, ISBN 978-1484236178
2. "Understanding Virtual Reality: Interface, Application and Design", William R Sherman and Alan B Craig, (The Morgan Kaufmann Series in Computer Graphics)". Morgan Kaufmann Publishers, San Francisco, CA, 2002
3. "Developing Virtual Reality Applications: Foundations of Effective Design", Alan B Craig, William R Sherman and Jeffrey D Will, Morgan Kaufmann, 2009
4. "Designing for Mixed Reality", Kharis O'Connell, O'Reilly Media, Inc., 2016, ISBN:9781491962381
5. "Theory and applications of marker-based augmented reality", Sanni Siltanen, Julkaisija, Utgivare Publisher. 2012. ISBN 978-951-38-7449-0
6. "Designing Virtual Systems: The Structured Approach", Gerard Jounghyun Kim, 2005

IV Year I Semester	INTERNET OF THINGS (PROFESSIONAL ELECTIVE-IV)	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of the course are to make student learn the application areas of IOT , the revolution of Internet in Mobile Devices, Cloud & Sensor Networks and building blocks of Internet of Things and characteristics

UNIT I:

Predecessors of IoT: Introduction, Wireless Sensor Networks, Machine-to-Machine Communications, Cyber Physical Systems

Emergence of IoT: Introduction, Evolution of IoT, Enabling IoT and the Complex Interdependence of Technologies, IoT Networking Components, Addressing Strategies in IoT

UNIT II:

IoT Sensing and Actuation: Introduction, Sensors, Sensor Characteristics, Sensorial Deviations, Sensing Types, Sensing Considerations, Actuators, Actuator Types, Actuator Characteristics

IoT Processing Topologies and Types: Data Format, Importance of Processing in IoT, Processing Topologies, IoT Device Design and Selection Considerations, Processing Offloading.

UNIT III:

IoT Connectivity Technologies: Introduction, IEEE 802.15.4, Zigbee, Thread, ISA100.11A, Wireless HART, RFID, NFC,DASH7, Z-Wave, Weightless, Sigfox, LoRa, NB-IT, Wi-Fi, Bluetooth

IoT Communication Technologies: Introduction, Infrastructure Protocols, Discovery Protocols, Data Protocols, Identification Protocols, Device Management, Semantic Protocols.

UNIT IV:

IoT Interoperability: Introduction, Standards, Frameworks

Fog Computing and Its Applications: Introduction, View of Fog Computing Architecture, Fog Computing in IoT, Selected Applications of Fog Computing

UNIT V:

Paradigms, Challenges, and the Future: Introduction, Evolution of New IoT Paradigms, Challenges Associated with IoT, Emerging Pillars of IoT

IoT Case Studies: Agricultural IoT, Vehicular IoT

Text Books:

1. Introduction to IoT, Sudip Misra, Anandarup Mukhaerjee, Arjit Roy, Cambridge University Press, 2021
2. Internet of Things: Architecture, Design Principles and Applications, Rajkamal, McGraw Hill Higher Education

Reference Books:

1. Fog and Edge Computing: Principles and Paradigms, Rajkumar Buyya (Editor), Satish Narayana Srirama (Editor) , ISBN: 978-1-119-52498-4, January 2019
2. Getting Started with the Internet of Things, Cuno Pfister , Oreilly

IV Year I Semester	AGENTIC AI (PROFESSIONAL ELECTIVE-IV)	L	T	P	C
		3	0	0	3

Course Outcomes:

CO1: Explain the fundamental concepts, characteristics, and architectures of Agentic AI systems, and differentiate them from traditional, reactive, and generative AI models.

CO2: Apply decision-making, planning, and reinforcement learning techniques to design autonomous agents capable of goal-directed behavior under deterministic and uncertain environments.

CO3: Analyze the role of Large Language Models (LLMs) in agentic systems and implement prompt-driven reasoning, memory, and tool-use mechanisms to enable intelligent agent behavior.

CO4: Design and develop single-agent and multi-agent systems using contemporary agentic AI frameworks, incorporating coordination, communication, and workflow orchestration.

CO5: Evaluate real-world applications of Agentic AI with respect to ethics, safety, responsibility, and societal impact, and propose value-aligned and secure agentic solutions.

UNIT- I — Foundations of Agentic Intelligence: Understand what Agentic AI is, how it differs from traditional and generative AI, and learn its core theoretical building blocks.
Wat is Agentic AI vs. Generative/Reactive AI; Autonomy, Proactivity, and Goal-Directed Behavior; Components of an intelligent agent: Perception, Reasoning, Learning, Action; Agent architectures: Reactive, Deliberative, Hybrid; Rationality and decision models; Real-world use cases and motivation for agentic systems

Unit II — Decision Making, Planning & Learning: Teach how agents make decisions, plan under uncertainty, and adapt through learning. Decision processes: utility theory, Markov Decision Processes (MDPs); Planning under uncertainty; Reinforcement learning fundamentals (MDP, policy, reward shaping); Deep RL essentials (PPO, DQN, A3C — as applicable); Meta-cognition and adaptation in agents; Ethical and value-aligned decision making

Unit III — Large Language Models & Prompt-Driven Agent Behavior: Cover how modern agentic systems leverage large language models and advanced prompt techniques.
Transformer architecture and attention basics; Role of LLMs in agentic systems; Prompt engineering fundamentals (zero-shot, chain-of-thought, ReAct); Retrieval-Augmented Generation (RAG) and memory systems; Context management and embedding techniques; Tool-usage and action invocation via prompts

UNIT- IV — Tools, Frameworks & System Design: Introduce hands-on frameworks used to build agentic AI and how to compose multiple agents into work flows.Overview of agent frameworks: LangChain, LangGraph, AutoGen, CrewAI (documentation & practice); Memory, embedding databases (e.g., FAISS, Pinecone, ChromaDB); Workflow orchestration and multi-agent coordination; Agent-to-agent communication; Deployment patterns and APIs; Monitoring, scalability, and production concerns

UNIT- V — Responsible & Applied Agentic AI: Address real-world applications, ethical considerations, safety, and career perspectives. Responsible agentic AI design and risk mitigation; Bias, fairness, and interpretability in autonomous systems; Privacy, transparency, and accountability. Safety and alignment challenges; Case studies: Healthcare, cybersecurity, business automation

Textbooks:
1. Stuart J. Russell and Peter Norvig, Artificial Intelligence: A Modern Approach (4th Ed.), Pearson.
2. Michael Wooldridge, An Introduction to MultiAgent Systems (2nd ed.)
3. Richard S. Sutton & Andrew G. Barto — Reinforcement Learning: An Introduction

References:
1. David L. Poole & Alan K. Mackworth — Artificial Intelligence: Foundations of Computational Agents
2. Ken Huang (Ed.) — Agentic AI: Theories and Practices (Springer, 2025)

IV Year I Semester	AGILE METHODOLOGIES (PROFESSIONAL ELECTIVE-V)	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of this course are to
- Introduce the important concepts of Agile software development Process
- Emphasize the role of stand-up meetings in software collaboration
- Impart the knowledge on values and principles in understanding agility

UNIT I :

Learning Agile: Agile, Getting Agile into your brain, Understanding Agile values, No Silver Bullet, Agile to the Rescue. A fractured perspective, The Agile Manifesto, Understanding the Elephant, Where to Start with a New Methodology.

UNIT II :

The Agile Principles: The 12 Principles of Agile Software, The Customer Is Always Right, Delivering the Project, Better Project Delivery for the Ebook Reader Project. Communicating and Working Together, Project Execution—Moving the Project Along, Constantly Improving the Project and the Team. The Agile Project: Bringing All the Principles Together

UNIT III :

SCRUM and Self-Organizing Teams: The Rules of Scrum, Act I: I Can Haz Scrum, Everyone on a Scrum Team owns the Project, Status Updates Are for Social Networks!, The Whole Team Uses the Daily Scrum, Feedback and the Visibility-Inspection-Adaptation Cycle, The Last Responsible Moment, Sprinting into a Wall, Sprints, Planning, and Retrospectives.

Scrum Planning And Collective Commitment: Not Quite Expecting the Unexpected, User Stories, Velocity, and Generally Accepted Scrum Practices, Victory Lap, Scrum Values Revisited.

UNIT IV :

XP And Embracing Change: Going into Overtime, The Primary Practices of XP, The Game Plan Changed, but We're Still Losing, The XP Values Help the Team Change Their Mindset, An Effective Mindset Starts with the XP Values, The Momentum Shifts, Understanding the XP Principles Helps You Embrace Change.

XP, Simplicity, and Incremental Design: Code and Design, Make Code and Design Decisions at the Last Responsible Moment, Final Score.

UNIT V:

Lean, Eliminating Waste, and Seeing the whole: Lean Thinking, Creating Heroes and Magical Thinking. Eliminate Waste, Gain a Deeper Understanding of the Product, Deliver As Fast As Possible.

Kanban, Flow, and Constantly Improving: The Principles of Kanban, Improving Your Process with Kanban, Measure and Manage Flow, Emergent Behavior with Kanban.
The Agile Coach: Coaches Understand Why People Don't Always Want to Change. The Principles of Coaching.

Text Books :
1. Andrew Stellman, Jill Alison Hart, Learning Agile, O'Reilly, 2015.

Reference Books:
1. Andrew Stellman, Jennifer Green, Head first Agile, O'Reilly, 2017.
2. Rubin K , Essential Scrum : A Practical Guide To The Most Popular Agile Process, Addison-Wesley, 2013

		L	T	P	C
IV Year I Semester	**GENERATIVE AI (PROFESSIONAL ELECTIVE-V)**	3	0	0	3

Course Objectives:

The objectives of the course are to understand the basics of Generative AI, Text Generation, the process of generating videos, GAN and its variants.

UNIT I :

Introduction To Gen Ai: Historical Overview of Generative modeling, Difference between Gen AI and Discriminative Modeling, Importance of generative models in AI and Machine Learning, Types of Generative models, GANs, VAEs, autoregressive models and Vector quantized Diffusion models, Understanding if probabilistic modeling and generative process, Challenges of Generative Modeling, Future of Gen AI, Ethical Aspects of AI, Responsible AI, Use Cases.

UNIT II:

Generative Models For Text: Language Models Basics, Building blocks of Language models, Transformer Architecture, Encoder and Decoder, Attention mechanisms, Generation of Text, Models like BERT and GPT models, Generation of Text, Autoencoding, Regression Models, Exploring ChatGPT, Prompt Engineering: Designing Prompts, Revising Prompts using Reinforcement Learning from Human Feedback (RLHF), Retrieval Augmented Generation, Multimodal LLM, Issues of LLM like hallucination.

UNIT III:

Generation of Images: Introduction to Generative Adversarial Networks, Adversarial Training Process, Nash Equilibrium, Variational Autoencoders, Encoder-Decoder Architectures, Stable Diffusion Models, Introduction to Transformer-based Image Generation, CLIP, Visual Transformers ViT- Dall-E2 and Dall-E3, GPT-4V, Issues of Image Generation models like Mode Collapse and Stability.

UNIT IV:

Generation of Painting, Music, and Play: Variants of GAN, Types of GAN, Cyclic GAN, Using Cyclic GAN to Generate Paintings, Neural Style Transfer, Style Transfer, Music Generating RNN, MuseGAN, Autonomous agents, Deep Q Algorithm, Actor-critic Network.

UNIT V:

Open Source Models And Programming Frameworks: Training and Fine tuning of Generative models, GPT 4 All, Transfer learning and Pretrained models, Training vision models, Google Copilot, Programming LLM, LangChain, Open Source Models, Llama, Programming for TimeSformer, Deployment, Hugging Face.

Text Books:
1. Denis Rothman, "Transformers for Natural Language Processing and Computer Vision", Third Edition , Packt Books, 2024

Reference Books:
1. David Foster, "Generative Deep Learning", O'Reily Books, 2024.
2. Altaf Rehmani, "Generative AI for Everyone", BlueRose One, 2024.

IV Year I Semester	COMPUTER VISION (PROFESSIONAL ELECTIVE-V)	L	T	P	C
		3	0	0	3

Course Objectives:
The objectives of the course are to understand the Fundamental Concepts related to sources, shadows and shading, the Geometry of Multiple Views

UNIT –I:
CAMERAS: Pinhole Cameras Radiometry – Measuring Light: Light in Space, Light Surfaces, Important Special Cases Sources, Shadows, And Shading: Qualitative Radiometry, Sources and Their Effects, Local Shading Models, Application: Photometric Stereo, Interreflections: Global Shading Models Color: The Physics of Color, Human Color Perception, Representing Color, A Model for Image Color, Surface Color from Image Color.

UNIT-II:
Linear Filters: Linear Filters and Convolution, Shift Invariant Linear Systems, Spatial Frequency and Fourier Transforms, Sampling and Aliasing, Filters as Templates, Edge Detection: Noise, Estimating Derivatives, Detecting Edges Texture0: Representing Texture, Analysis (and Synthesis) Using Oriented Pyramids, Application: Synthesis by Sampling Local Models, Shape from Texture.

UNIT-III:
The Geometry of Multiple Views: Two Views Stereopsis: Reconstruction, Human Stereopsis, Binocular Fusion, Using More Cameras Segmentation by Clustering: What Is Segmentation? Human Vision: Grouping and Getstalt, Applications: Shot Boundary Detection and Background Subtraction, Image Segmentation by Clustering Pixels, Segmentation by Graph-Theoretic Clustering,

UNIT-IV:
Segmentation by Fitting a Model: The Hough Transform, Fitting Lines, Fitting Curves, Fitting as a Probabilistic Inference Problem, Robustness Segmentation and Fitting Using Probabilistic Methods: Missing Data Problems, Fitting, and Segmentation, The EM Algorithm in Practice, Tracking With Linear Dynamic Models: Tracking as an Abstract Inference Problem, Linear Dynamic Models, Kalman Filtering, Data Association, Applications and Examples

UNIT- V:
Geometric Camera Models: Elements of Analytical Euclidean Geometry, Camera Parameters and the Perspective Projection, Affine Cameras and Affine Projection Equations Geometric Camera Calibration: Least-Squares Parameter Estimation, A Linear Approach to Camera Calibration, Taking Radial Distortion into Account, Analytical Photogrammetry,
Case study: Mobile Robot Localization Model- Based Vision: Initial Assumptions, Obtaining Hypotheses by Pose Consistency, Obtaining Hypotheses by pose Clustering, Obtaining Hypotheses Using Invariants, Verification, Case study: Registration in Medical Imaging Systems, Curved Surfaces and Alignment.

Text Books:
1. David A. Forsyth, Jean Ponce, "Computer Vision – A Modern Approach", PHI Learning (Indian Edition), 2009.

Reference Books:
1. E. R. Davies, "Computer and Machine Vision – Theory, Algorithms and Practicalities", Elsevier (Academic Press), 4th edition, 2013.
2. R. C. Gonzalez, R. E. Woods, "Digital Image Processing", Addison Wesley, 2008.
3. Richard Szeliski "Computer Vision: Algorithms and Applications" Springer, Verlag London Limited, 2011.

IV Year I Semester	CYBER PHYSICAL SYSTEMS (PROFESSIONAL ELECTIVE-V)	L	T	P	C
		3	0	0	3

Course Objectives:
The main objectives of the course are to
- Discuss the core principles behind Cyber Physical Systems
- Identify Security mechanisms of Cyber Physical System
- Describe Synchronization in Distributed Cyber-Physical Systems

UNIT I:
Symbolic Synthesis for Cyber-Physical Systems: Introduction and Motivation, Basic Techniques - Preliminaries, Problem Definition, Solving the Synthesis Problem, Construction of Symbolic Models, Advanced Techniques: Construction of Symbolic Models, Continuous-Time Controllers, Software Tools

UNIT II:
Security of Cyber-Physical Systems: Introduction and Motivation, Basic Techniques - Cyber Security Requirements, Attack Model, Countermeasures, Advanced Techniques: System Theoretic Approaches

UNIT III:
Synchronization in Distributed Cyber-Physical Systems: Challenges in Cyber-Physical Systems, A Complexity-Reducing Technique for Synchronization, Formal Software Engineering, Distributed Consensus Algorithms, Synchronous Lockstep Executions, Time-Triggered Architecture, Related Technology, Advanced Techniques

UNIT IV:
Real-Time Scheduling for Cyber-Physical Systems: Introduction and Motivation, Basic Techniques, Scheduling with Fixed Timing Parameters, Memory Effects, Multiprocessor/ Multicore Scheduling, Accommodating Variability and Uncertainty

UNIT V:
Model Integration in Cyber-Physical Systems: Introduction and Motivation, Causality, Semantic Domains for Time, Interaction Models for Computational Processes, Semantics of CPS DSMLs, Advanced Techniques, ForSpec, The Syntax of CyPhyML, Formalization of Semantics, Formalization of Language Integration.

Text Books:
1. Raj Kumar, Dionisio De Niz, Mark Klein, Cyber-Physical Systems, Addison-Wesley Professional, 2016
2. Rajeev Alur, Principles of Cyber-Physical Systems, MIT Press

Reference Books:
1. E.A.Lee, Sanjit Seshia, Introduction to Embedded Systems: A Cyber-Physical Systems Approach, MIT Press
2. Andre Platzer, Logical Foundations of Cyber-Physical Systems, 2e, Springer Publishing, 2018

		L	T	P	C
IV Year I Semester	**PROMPT ENGINEERING (SKILL ENHANCEMENT COURSE)**	0	1	2	2

Course Objectives:
The main objectives of the course are to
- Apply iterative prompting for clarity and context.
- Create varied prompts to steer model outputs.
- Construct chain-of-thought and structured prompts.
- Develop retrieval-augmented pipelines to ground outputs.
- Evaluate LLM agents and multimodal apps for ethics and robustness.

Unit I: Foundations of Prompt Engineering: Definition of prompt engineering, Distinction between prompt engineering and model fine-tuning, Motivation and benefits of prompt engineering, Core principles of effective prompt design, Anatomy of a prompt, Setting up the Python environment for LLM interaction, Iterative prompting lifecycle, Common prompt pitfalls and remediation

Lab Experiments:
1. Environment & Connectivity: Install required packages (e.g., transformers, openai); securely configure the API key; run a simple "Hello, world" prompt to verify model access.
2. Baseline vs. Enhanced Prompts: Execute a naïve prompt ("Write a one-paragraph bio of Ada Lovelace.") and an enhanced prompt that adds role framing, specificity, and explicit format instructions; compare both outputs for relevance, completeness, and style.
3. Iterative Refinement on a Simple Task: Summarize the plot of the Shakespearean play Romeo and Juliet in two sentences through three rounds of prompt tweaking:
    a. Minimal instruction.
    b. Addition of length and style constraints
    c. Specification of key content elements (setting and theme)
    Document how each iteration changes and improves the result.
4. Diagnosing Prompt Failures & Edge Cases: Craft a vague or contradictory prompt; analyze the failure mode (ambiguity, missing context, or format errors); refine the prompt by adding examples or clarifying instructions.

Unit II: Advanced Prompt Patterns & Techniques: Enhanced prompt anatomy: contextual detail and explicit output specifications, Few-shot in-context prompting, Prompt structuring and template design, Role-based prompting to establish personas or system behavior, Negative prompting to filter or suppress undesired content, Constraint specification and instruction enforcement (e.g., length, format), Iterative prompt refinement and optimization

Lab Experiments:
1. Few-Shot vs. Zero-Shot Comparison: Design and execute a zero-shot prompt and a few-shot prompt (with 2–3 exemplar input-output pairs) for a chosen text task (e.g., sentiment classification or translation); compare outputs for accuracy, consistency, and adherence to examples.
2. Role-Based & Negative Prompting: Craft a role-based prompt to establish a specific persona (e.g., "You are a financial advisor…"); then create a negative prompt to suppress undesired content (e.g., "Do not mention any brand names"); evaluate how each influences the model's response.

3. Constraint Specification & Iterative Refinement: Select an open-ended task (e.g., summarizing a technical article); issue a basic prompt; identify failures in length or format; refine the prompt by adding explicit constraints (word count, bullet format, etc.); document improvements over two refinement cycles.

Unit III: Structured Output & Reasoning Techniques: Importance of structured outputs for real-world applications, Prompting for specific formats (lists, tables, Markdown), Generating valid JSON and YAML via explicit instructions, Eliciting chain-of-thought reasoning in zero-shot prompts, Decomposing complex tasks into manageable sub-tasks

Lab Experiments:
1. Structured Format Prompting: Instruct the model to output information as bullet lists and Markdown tables (e.g., "List three benefits of daily exercise in a Markdown table with columns 'Benefit' and 'Description.'"); verify the output matches the requested structure.
2. JSON/YAML Generation: Provide a brief dataset description (e.g., three books with title, author, publication year) and prompt the model to produce valid JSON or YAML; use a parser to validate syntax and refine the prompt if errors occur.
3. Chain-of-Thought & Task Decomposition: Present a multi-step problem (e.g., a logic puzzle) and apply zero-shot CoT prompting (e.g., "Let's think step by step. Explain your reasoning before the final answer."); separately, decompose the problem into sequential sub-questions, collect partial answers, combine them, and compare accuracy against a direct-answer baseline.

Unit IV: Retrieval-Augmented Generation & LangChain Workflows: Limitations of LLM internal knowledge, Need for external data sources, Introduction to Retrieval-Augmented Generation (RAG), Overview of RAG architecture (indexing vs. retrieval + generation), Getting started with LangChain for LLM applications, Basics of LangChain Expression Language (LCEL), Simplified indexing pipeline: document loading & text splitting, Fundamentals of embeddings and vector stores, Building a basic retrieval-generation pipeline with an LCEL chain

Lab Experiments:
1. Building a Simple LCEL Chain: Create a minimal LCEL script that accepts a fixed instruction (e.g., "Summarize this text: …"), passes it to an LLM, and prints the result; verify end-to-end execution.
2. Basic Data Indexing for RAG: Load a small collection of documents; split into uniform chunks (e.g., 200 tokens); generate embeddings for each chunk; store them in an in-memory vector store; inspect for consistency.
3. Constructing & Running a Basic RAG Chain: Build a pipeline that:
   a. Receives a user query
   b. Retrieves the top-k relevant chunks
   c. Constructs a combined prompt with context + query
   d. Send it to the LLM
   e. Returns the answer

Test with sample queries and compare factual accuracy against a prompt without retrieval.

Unit V: Agents, Multimodal AI & Ethical Evaluation: Introduction to LLM agents and their basic architecture, Overview of multimodal AI models (VLMs), Prompting for text-to-image generation and image understanding, Importance of prompt evaluation beyond

subjective judgment, Manual evaluation techniques (heuristic checks for accuracy, relevance, format), Introduction to "LLM-as-Judge" for automated evaluation, Security considerations (prompt injection, sensitive-information risks), Prompt-based mitigation strategies for safety and robustness, Ethical concerns (bias, misinformation, data privacy), Brief exploration of UI frameworks (Streamlit/Gradio) for deploying prompt-driven apps, Adapting to the evolving nature of prompt engineering through continuous learning

Lab Experiments:

1. Building a Simple LLM Agent: Register a tool (e.g., a calculator function) and craft prompts that instruct the agent to invoke it when required; implement using LangChain or a function-calling API; test on queries requiring tool execution.

2. Multimodal Prompting Exploration: Generate images from detailed text prompts; feed one generated image into an image-understanding model or API with an appropriate prompt; compare the returned caption to the original prompt to evaluate alignment.

3. Prompt Evaluation & Ethics Workshop:
   a. Select two existing prompts and generate multiple outputs; apply manual heuristic checks for accuracy, relevance, and format compliance.
   b. Use an "LLM-as-Judge" prompt (e.g., "Rate these outputs on a scale of 1–5 for clarity and correctness.") to automate evaluation.
   c. Design a prompt- injection test (e.g., "Ignore previous instructions…"), observe the response, then refine system prompts to mitigate the vulnerability.

IV Year I Semester	CONSTITUTION OF INDIA	L	T	P	C
		2	0	0	-

Course Objectives:
The objectives of the course are to
- Understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.
- Address the growth of Indian opinion regarding modern Indian intellectuals' constitutional role and entitlement to civil and economic rights as well as the emergence of nationhood in the early years of Indian nationalism.
- Address the role of socialism in India after the commencement of the Bolshevik Revolution in 1917 and its impact on the initial drafting of the Indian Constitution.

UNIT–I:
History of Making of the Indian Constitution: History, Drafting Committee, (Composition & Working)
Philosophy of the Indian Constitution- Preamble, Salient, Features

UNIT–II:
Contours of Constitutional Rights & Duties: Fundamental Rights, Right to Equality, Right to Freedom, Right against Exploitation, Right to Freedom of Religion, Cultural and Educational Rights, Right to Constitutional Remedies, Directive Principles of State Policy, Fundamental Duties.

UNIT–III:
Organs of Governance: Parliament, Composition, Qualifications and Disqualifications, Powers and Functions, **Executive-** President, Governor, Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualifications, Powers and Functions

UNIT–IV:
Local Administration: District's Administration head: Role and Importance, Municipalities: Introduction, Mayor and role of Elected Representative CEO of Municipal Corporation, Pachayati raj: Introduction, PRI: Zila Pachayat, Elected officials and their roles, CEO Zila Pachayat: Position and role, Block level: Organizational Hierarchy (Different departments), Village level: Role of Elected and Appointed officials, Importance of grass root democracy

UNIT–V:
Election Commission: Election Commission: Role and Functioning, Chief Election Commissioner and Election Commissioners, State Election Commission: Role and Functioning, Institute and Bodies for the welfare of SC/ST/OBC and women.

Text Books:
1. The Constitution of India, 1st Edition, (Bare Act), Government Publication, 1950
2. Framing of Indian Constitution, 1st Edition, Dr. S. N. Busi, Dr. B. R. Ambedkar, 2015

Reference Books:
1. Indian Constitution Law, 7th Edition, M. P. Jain, Lexis Nexis, 2014
   Rubrics for Project Work:

B.Tech. – IV Year II Semester

S.No.		Category	Title	L	T	P	Credits
1		Internship & Project Work	Full semester Internship &Project Work	0	0	24	12