B. Tech – II Year I Semester

S.No	Category	Title	L	T	P	C
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	Artificial Intelligence	3	0	0	3
4	Professional Core	Advanced Data Structures & Algorithms Analysis	3	0	0	3
5	Professional Core	Object-Oriented Programming Through JAVA	3	0	0	3
6	Professional Core	Advanced Data Structures and Algorithms 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			**15**	**2**	**10**	**20**

B. Tech– II Year II Semester

S.No	Category	Title	L	T	P	C
1	Management Course- I	Optimization Techniques	2	0	0	2
2	Engineering Science/ Basic Science	Probability & Statistics	3	0	0	3
3	Professional Core	Machine Learning	3	0	0	3
4	Professional Core	Database Management Systems	3	0	0	3
5	Professional Core	Digital Logic and Computer Organization	3	0	0	3
6	Professional Core	AI & ML 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-1	0	1	2	2
9	BS&H	Design Thinking & Innovation	1	0	2	2
Total			**15**	**1**	**12**	**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 students will be able to

1. Build skills in solving mathematical problems (L3)
2. Comprehend mathematical principles and logic (L4)
3. Demonstrate knowledge of mathematical modeling and proficiency in using mathematical software (L6)
4. Manipulate and analyze data numerically and/or graphicallysing appropriate Software (L3)
5. How to communicate effectively mathematical ideas/results verbally or in writing (L1)

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 ofProof, 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:

- Define the terms like Natural Acceptance, Happiness and Prosperity (L1, L2)
- Identify one's self, and one's surroundings (family, society nature) (L1, L2)
- Apply what they have learnt to their own self in different day-to-day settings in real life (L3)
- Relate human values with human relationship and human society. (L4)
- Justify the need for universal human values and harmonious existence (L5)
- 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 session)   Harmony in the Nature/Existence (4 lectures and 2 tutorials for practice

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%2023-25%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

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

II Year I Semester

L	T	P	C
3	0	0	3

ARTIFICIAL INTELLIGENCE

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

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

II Year I Semester

L	T	P	C
3	0	0	3

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

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/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, 1. Introduction to Algorithms (youtube.com)

II Year I Semester

L	T	P	C
3	0	0	3

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

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, ResultSet 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, 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

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

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 bi-connected 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

II Year I Semester

L	T	P	C
0	0	3	1.5

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

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, JavaFX 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 StringBuffer 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

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

● 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

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

2. Without writing any code, build a GUI that display text in label and image in an ImageView (use JavaFX)

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

Exercise – 9

1. 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

II Year I Semester

L	T	P	C
0	1	2	2

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

UNTI-I: History of Python Programming Language, Thrust Areas of Python, Installing Anaconda Python Distribution, Installing and Using Jupyter 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 Operators viii) 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, 2ndEdition, 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

ENVIRONMENTAL SCIENCE

L	T	P	C
2	0	0	0

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:
- Grasp multidisciplinary nature of environmental studies and various renewable and non-renewable resources.
- Understand flow and bio-geo-chemical cycles and ecological pyramids.
- Understand various causes of pollution and solid waste management and related preventive measures.
- About the rainwater harvesting, watershed management, ozone layer depletion and waste land reclamation.
- Casus of population explosion, value education and 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, casestudies.–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-situ and 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 land slides.

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.RaghavanNambiar,"Text book of Environmental Studies for Undergraduate Courses as per UGC model syllabus", Scitech Publications (India), Pvt.Ltd.

Reference Books:
1. DeekshaDaveandE.SaiBabaReddy, "Text book of Environmental Science",Cengage Publications.
2. M.AnjiReddy,"Text book of Environmental Sciences and Technology",BSPublication.
3. J.P.Sharma,Comprehensive Environmental studies,Laxmi publications.
4. J.GlynnHenryandGaryW.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 WendellP.Ela,"Introduction to Environmental Engineering and Science,Prentice Hall of India Private limited.

II Year II Semester

L	T	P	C
3	0	0	3

OPTIMIZATION TECHNIQUES

Pre-requisite:

Course Objectives:
1. To define an objective function and constraint functions in terms of design variables, and then state the optimization problem.
2. To state single variable and multi variable optimization problems, without and with constraints.
3. To explain linear programming technique to an optimization problem, define slack and surplus variables, by using Simplex method.
4. To state transportation and assignment problem as a linear programming problem to determine Simplex method.
5. To study and explain nonlinear programming techniques, unconstrained or constrained, and define exterior and interior penalty functions for optimization problems.

Course Outcomes: At the end of the course, student will be able to
- State and formulate the optimization problem, without and with constraints, by using design variables from an engineering design problem.
- Apply classical optimization techniques to minimize or maximize a multi-variable objective function, without or with constraints, and arrive at an optimal solution.
- Apply and Solve transportation and assignment problem by using Linear programming Simplex method.
- Apply gradient and non-gradient methods to nonlinear optimization problems and use interior or exterior penalty functions for the constraints to derive the optimal solutions
- Formulate and apply Dynamic programming technique to inventory control, production planning, engineering design problems etc. to reach a final optimal solution from the current optimal solution.

UNIT I: Introduction and Classical Optimization Techniques:
Statement of an Optimization problem, design vector, design constraints, constraint surface, objective function, objective function surfaces, classification of Optimization problems.
Classical Optimization Techniques: Single variable Optimization, multi variable Optimization without constraints, necessary and sufficient conditions for minimum/maximum, multivariable Optimization with equality constraints. Solution by method of Lagrange multipliers, multivariable Optimization with inequality constraints, Kuhn – Tucker conditions

UNIT II: Linear Programming :
Standard form of a linear programming problem, geometry of linear programming problems, definitions and theorems, solution of a system of linear simultaneous equations, pivotal reduction of a general system of equations, motivation to the simplex method, simplex algorithm.

UNIT III: Transportation Problem:
Finding initial basic feasible solution by north – west corner rule, least cost method and Vogel's approximation method, testing for optimality of balanced transportation problems, Special cases in transportation problem.

UNIT IV: Nonlinear Programming:
Unconstrained cases, One – dimensional minimization methods: Classification, Fibonacci method, Univariate method, steepest descent method. Constrained cases– Characteristics of a constrained problem, Classification, Basic approach of Penalty Function method, Basic approaches of Interior and Exterior penalty function methods,

UNIT V: Dynamic Programming:
Dynamic programming multistage decision processes, types, concept of sub optimization and the principle of optimality, computational procedure in dynamic programming, examples illustrating the calculus method of solution, examples illustrating the tabular method of solution.

Textbooks:
1. "Engineering optimization: Theory and practice", S. S.Rao, New Age International (P) Limited, 3rd edition, 1998.
2. "Introductory Operations Research", H.S. Kasene& K.D. Kumar, Springer (India), Pvt.LTd.

Reference Books:
1. "Optimization Methods in Operations Research and systems Analysis", by K.V. Mital and C. Mohan, New Age International (P) Limited, Publishers, 3rd edition, 1996.
2. Operations Research, Dr.S.D.Sharma, Kedarnath, Ramnath& Co

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

MACHINE LEARNING

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.

II Year II Semester

L	T	P	C
3	0	0	3

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

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_0127580666728202 2456_shared/overview

II Year II 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

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, 2023.
2. Digital Design, 6th Edition, M. Morris Mano, Pearson Education, 2018.
3. Computer Organization and Architecture, William Stallings, 11thEdition, Pearson, 2022.

Reference Books:
1. Computer Systems Architecture, M.Moris Mano, 3rdEdition, Pearson, 2017.
2. Computer Organization and Design, David A. Paterson, John L. Hennessy, Elsevier, 2004.
3. Fundamentals of Logic Design, Roth, 5thEdition, Thomson, 2003.

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

II Year II Semester

L	T	P	C
0	0	3	1.5

AI & ML LAB

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 and machine learning.
4. To learn about computing central tendency measures and Data preprocessing techniques
5. To learn about classification and regression algorithms
6. To apply different clustering algorithms for a problem.

Software Required for ML: Python/R/Weka

List of Experiments
1. Pandas Library
   a) Write a python program to implement Pandas Series with labels.
   b) Create a Pandas Series from a dictionary.
   c) Creating a Pandas Data Frame.
   d) Write a program which makes use of the following Pandas methods
      i) describe ()   ii) head ()   iii) tail ()   iv) info ()
2. Pandas Library: Visualization
   a) Write a program which use pandas inbuilt visualization to plot following graphs:
      i. Bar plots   ii. Histograms   iii. Line plots   iv. Scatter plots
3. Write a Program to Implement Breadth First Search using Python.
4. Write a program to implement Best First Searching Algorithm
5. Write a Program to Implement Depth First Search using Python.
6. Write a program to implement the Heuristic Search
7. Write a python program to implement A* and AO* algorithm. (Ex: find the shortest path)
8. Apply the following Pre-processing techniques for a given dataset.
   a. Attribute selection
   b. Handling Missing Values
   c. Discretization
   d. Elimination of Outliers
9. Apply KNN algorithm for classification and regression
10. Demonstrate decision tree algorithm for a classification problem and perform parameter tuning for better results
11. Apply Random Forest algorithm for classification and regression
12. Demonstrate Naïve Bayes Classification algorithm.
13. Apply Support Vector algorithm for classification
14. Implement the K-means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of the Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameters K.

II Year II Semester

L	T	P	C
0	0	3	1.5

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

II Year II Semester

L	T	P	C
0	1	2	2

FULL STACK DEVELOPMENT – 1
(SKILL ENHANCEMENT COURSE)

Course Objectives:

The main objectives of the course are to

1. Make use of HTML elements and their attributes for designing static web pages
2. Build a web page by applying appropriate CSS styles to HTML elements
3. 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

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

- 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.).
- 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).
- 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
   - Simple selector (element, id, class, group, universal)
   - Combinator selector (descendant, child, adjacent sibling, general sibling)
   - Pseudo-class selector
   - Pseudo-element selector
   - 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. JavaScript 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. JavaScript 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. Javascript Functions and Events

a. Design a appropriate function should be called to display
   ● Factorial of that number
   ● Fibonacci series up to that number
   ● Prime numbers up to that number
   ● 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
   4. Factorial of that number
   5. Fibonacci series up to that number
   6. Prime numbers up to that number
   7. 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
5. https://www.w3schools.com/typescript

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	Blooms 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	Deep Learning		3	0	0	3
2	Professional Core	Computer Networks		3	0	0	3
3	Professional Core	Operating Systems		3	0	0	3
4	Professional Elective-I	1. Object Oriented Analysis & Design 2. Automata Theory and Compiler Design 3. Soft computing 4. Internet of Things		3	0	0	3
5	Open Elective-I	OR Entrepreneurship Development & Venture Creation		3	0	0	3
6	Professional Core	Deep learning 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 / SWAYAM Plus – Data Engineer / AI Engineer /		0	1	2	2
9	ES	User Interface Design using Flutter / SWAYAM Plus - Android Application Development (with Flutter)		0	0	2	1
10	Evaluation of Community Service Project Internship			-	-	-	2
	Total			**15**	**1**	**10**	**23**
MC	Student may select from the Same Minor Pool			3	0	3	4.5
MC	Minor Course through SWAYAM / NPTEL (Minimum 12 Week, 3 credit course)			3	0	0	3
HC	Student may select from the Same Honors Pool			3	0	0	3
HC	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	Software Engineering	3	0	0	3
2	Professional Core	Generative A.I.	3	0	0	3
3	Professional Core	Data Visualization	3	0	0	3
4	Professional Elective-II	1. Software Testing Methodologies 2. Cryptography & Network Security 3. Recommender Systems 4. DevOps 5. Any of the 12-Week SWAYAM /NPTEL Course suggested by the BoS	3	0	0	3
5	Professional Elective-III	1. Software Project Management 2. Mobile Adhoc Networks 3. Computer Vision 4. Cloud Computing 5. Any of the 12-Week SWAYAM /NPTEL Course suggested by the BoS	3	0	0	3
6	Open Elective - II		3	0	0	3
7	Professional Core	Generative A.I. Lab	0	0	3	1.5
8	Professional Core	Data Visualization 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**	**8**	**23**
	*Mandatory Industry Internship/Mini Project of 08 weeks duration during summer vacation					
MC	Student may select from the same minors pool		3	0	3	4.5
MC	Minor Course (Student may select from the same specialized minors pool)		3	0	0	3
HC	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*

For Minor in AI:	L-T-P-C
1. Database Management Systems	3-0-3-4.5 (II-II)
2. Artificial Intelligence	3-0-0-3 (III-I)
3. Advanced Data Structures & Algorithm Analysis	3-0-3-4.5 (III-II)
4. Operating Systems	3-0-0-3 (IV-I)

Any of the following 12 Week 3 credit NPTEL MOOC Courses
5. Introduction to Machine Learning
6. Programming in Java
7. Computer Networks and Internet Protocol
8. Fundamentals of Object Oriented Programming
9. Discrete Mathematics for CS
10. Software Engineering
11. Deep Learning

<u>**Note**</u>: As per AICTE APH 2023-24 student can opt any one of the following streams:

1. Honors in CSE
2. Honors in AI&ML
3. Honors in Data Science

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
1. Deep Learning for Computer Vision (MOOCS- SWAYAM / NPTEL 12W)
2. Applied Linear Algebra in AI & ML (MOOCS- SWAYAM / NPTEL 12W)
3. Applied Time-Series Analysis (MOOCS- SWAYAM / NPTEL 12W)
4. Machine Learning for Engineering and Science Applications (MOOCS- SWAYAM / NPTEL 12W)
5. Practical High-Performance Computing (MOOCS- SWAYAM / NPTEL 12W)
6. Deep Learning for Natural Language Processing (MOOCS- SWAYAM / NPTEL 12W)
7. Privacy and Security in Online Social Media (MOOCS- SWAYAM / NPTEL 12W)
8. Natural Language Processing (MOOCS- SWAYAM / NPTEL 12W)

III Year I Semester	DEEP LEARNING	L	T	P	C
		3	0	0	3

Course Objectives:

The objective of this course is to cover the fundamentals of neural networks as well as some advanced topics such as recurrent neural networks, long short-term memory cells and convolution neural networks.

Course Outcomes:

After completion of course, students would be able to:
- Explore feed forward networks and Deep Neural networks
- Mathematically understand the deep learning approaches and paradigms
- Apply the deep learning techniques for various applications

UNIT-I:

Basics- Biological Neuron, Idea of computational units, McCulloch–Pitts unit and Thresholding logic, Linear Perceptron, Perceptron Learning Algorithm, Linear separability, Convergence theorem for Perceptron Learning Algorithm.

UNIT-II:

Feed forward Networks- Multilayer Perceptron, Gradient Descent, Back propagation, Empirical Risk Minimization, regularization, auto encoders. Deep Neural Networks: Difficulty of training deep neural networks, Greedylayer wise training.

UNIT-III:

Better Training of Neural Networks –Newer optimization methods for neural networks (Adagrad, adadelta, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization).

UNITIV:

Recurrent Neural Networks- Back propagation through time, Long Short-Term Memory, Gated Recurrent Units, Bidirectional LSTMs, Bidirectional RNNs.
Convolutional Neural Networks: LeNet, AlexNet. Generative models: Restrictive Boltzmann Machines (RBMs), Introduction to MCMC and Gibbs Sampling, gradient computations in RBMs, Deep Boltzmann Machines.

UNITV:

Recent trends-Variational Auto encoders,Transformers,GPT Applications: Vision, NLP, Speech

Text Books:

1. Deep Learning, Ian Good fellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016

Reference Books:

1. Neural Networks:A Systematic Introduction,RaúlRojas,1996
2. Pattern Recognition and Machine Learning, Christopher Bishop,2007
3. Deep Learning with Python, François Chollet, Manning Publications,2017

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.

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	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 dead lock 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: Multi threading 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, Dead lock characterization, Methods for handling Dead locks, Dead lock 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,GalvinPB,GagneG,10thEdition,Wiley, 2018.
2. Modern Operating Systems, TanenbaumAS,4thEdition ,Pearson ,2016.

Reference Books:

1. Operating Systems -Internals and Design Principles, Stallings W, 9thedition, Pearson, 2018
2. Operating Systems: A Concept Based Approach,D.MDhamdhere,3rdEdition, McGraw- Hill, 2013

Online Learning Resources:

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

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 withal 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.

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.

UNITIV:

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

UNITV:

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, BobbiJ. Young,Jim Conallen, KelliaHouston, "Object- Oriented Analysis and Design with Applications", 3rd edition,2013,PEARSON.
2. Grady Booch,James Rumbaugh, IvarJacobson:TheUnifiedModelingLanguageUserGuide,PearsonEducation.

Reference Books:

1. Meilir Page- Jones: Fundamentals of Object-Oriented Design in UML,Pearson Education.
2. Pascal Roques:Modeling Software Systems UsingUML2,WILEY- Dream tech India Pvt.Ltd.
3. AtulKahate: Object Oriented Analysis &Design, The McGraw-Hill Companies.

Appling UML and Patterns:AnintroductiontoObject–OrientedAnalysisandDesignandUnifiedProcess,CraigLarman,PearsonEducation.

III Year I Semester	AUTOMATA THEORY & COMPILER DESIGN	L	T	P	C
		3	0	0	3

Course Objectives:
- Introduce the notion of formal languages and grammars
- Design of Grammars, FAs and PDAs
- To become familiar with the underlying theory and methods used in compiler design
- To Introduce the parsing techniques, code optimization techniques and generate code

UNIT – I: Regular Expressions, Languages and Finite Automata
Formal Languages and the Chomsky Hierarchy, Regular Expressions and Regular Languages, Algebraic Laws for Regular Expressions, Applications of Regular Expressions, Abstract model of Finite Automaton, Transition Tables and Transition Graphs, Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA), Converting NFA to DFA, Finite Automata with ϵ transitions (NFA-ϵ), Converting NFA-ϵ to NFA/DFA, Minimization of Finite Automata, Equivalence of FA and Regular Expressions

UNIT-II: Context Free Grammars and Push Down Automata:
Context Free Grammars (CFG) and Context Free Languages (CFL), Design of CFGs, Leftmost and Rightmost Derivations, Parse Trees, Applications of CFGs, Ambiguity in Grammars and Languages, Push Down Automata (PDA), The Language of a PDA, Equivalence of PDAs and CFGs

UNIT-III: Lexical Analysis and Top-Down Parsing
The structure of a compiler, Role of lexical analyzer, Input Buffering, Specification of tokens, Recognition of tokens, The Lexical Analyser Generator –LEX
Introduction to Syntax Analysis, Eliminating ambiguity and left recursion from a CFG, Recursive Decent Parsing, LL(1) Grammars, Nonrecursive Predictive Parsing

UNIT-IV: Bottom-Up Parsing and Syntax Directed Translation
Shift-Reduce Parsing, Simple LR parsing, Canonical LR(1) Parsing, LALR Parsing, Parser Generators
Syntax Directed Definitions, Evaluation Orders for SDDs, Syntax Directed Translation Schemes

UNIT-V: Intermediate Code Generation, Code Generation and Optimization:
Three address code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Issues in the design of a Code Generator, The Target Language, A simple Code GeneratorBasic   Blocks and Flow Graphs, Optimization of Basic Blocks, Peephole Optimization

Textbooks:
1. Introduction to Automata Theory, Languages and Computation, J.E.Hopcroft, R.Motwani and J.D.Ullman, 3rd Edition, Pearson, 2008.
2. Compilers Principles, Techniques and Tools, 2nd Edition, Alfred V.Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson

Reference Books:
1. Introduction to Languages and The Theory of Computation, John C. Martin, McGraw Hill.
2. Theory of Computer Science-Automata, Languages and Computation, K.L.P.Mishra and N.Chandrasekaran, 3rd Edition, PHI, 2007
3. Compiler Construction‖, K.V.N. Sunitha, Pearson, 2013
4. Compiler Design, SandeepSaxena, Rajkumar Singh Rathore, S.Chand publication

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

Course Objectives:

To introduce the concepts in Soft Computing such as Artificial Neural Networks, Fuzzylogic-based systems, genetic algorithm-based systems and their hybrids.

Course Outcomes:

The students will be able to

- Learn soft computing techniques and their applications.
- Analyze various neural network architectures.
- Define the fuzzy systems.
- Understand the genetic algorithm concepts and their applications.
- Identify and select a suitable Soft Computing technology to solve the problem; construct a solution and implement a Soft Computing solution

UNIT-I:

Introduction to Soft Computing, Artificial neural networks, biological neurons, Basic models of artificial neural networks, Connections, Learning, Activation Functions, McCulloch and Pitts Neuron, Hebb network.

UNIT-II:

Perceptron networks, learning rule, Training and testing algorithm, Adaptive Linear Neuron, Back propagation Network, Architecture, Training algorithm

UNIT-III:

Fuzzy logic, fuzzy sets, properties, operations on fuzzy sets, fuzzy relations, operations on fuzzy relations, Fuzzy membership functions, fuzzification, Methods of membership, value assignments, intuition, inference, rank ordering, Lambda –
Cuts for fuzzy sets, Defuzzification methods

UNIT-IV:

Truth values and Tables in Fuzzy Logic, Fuzzy propositions, Formation of fuzzy rules, Decomposition of rules, Aggregation of rules, Fuzzy Inference Systems, Mamdani and Sugeno types, Neuro-fuzzy hybrid systems, characteristics, classification

UNIT-V:

Introduction to genetic algorithm, operators in genetic algorithm, coding, selection, crossover, mutation, stopping condition for genetic algorithm flow, Genetic-neuro hybrid systems, Genetic Fuzzy rule-based system

Text Books:

1. S.N.Sivanandam and S.N.Deepa, Principles of soft computing–JohnWiley & Sons,2007.
2. Timothy J.Ross,Fuzzy Logic with engineering applications, John Wiley &Sons, 2016.

Reference Books:

1. N.K.Sinha and M.M.Gupta, Soft Computing & Intelligent Systems:Theory & Applications-Academic Press /Elsevier. 2009.
2. Simon Haykin, Neural Network- A Comprehensive Foundation-Prentice Hall International, Inc.1998
3. R. Eberhart and Y. Shi, Computational Intelligence:
4. Concepts to Implementation, Morgan Kaufman/Elsevier, 2007.
5. Driankov D.,Hellendoorn H.and Reinfrank M. An Introduction to Fuzzy Control Narosa Pub., 2001.
6. BartKosko, Neural Network and Fuzzy Systems-Prentice Hall, Inc.,Englewood Cliffs, 1992
7. Goldberg D.E., Genetic Algorithms in Search, Optimization, and Machine Learning Addison Wesley, 1989

III Year I Semester	INTERNET OF THINGS	L	T	P	C
	(Common to AI&DS, AI&ML, CSE(AI), CSE (DS), CSE (AI&DS), CSE (AI&ML))	3	0	0	3

Course Objectives:
- Vision and Introduction to Internet of Things (IoT).
- Understand IoT Market perspective.
- Data and Knowledge Management and use of Devices in IoT Technology.
- Understand State of the Art – IoT Architecture.
- Understand Real World IoT Design Constraints, Industrial Automation and Commercial.

UNIT-I:
The Internet of Things: An Overview of Internet of things, Internet of Things Technology, behind IoTs Sources of the IoTs, M2M Communication, Examples of IoTs, Design Principles For Connected Devices Internet Connectivity Principles, Internet connectivity, Application Layer Protocols: HTTP, HTTPS, FTP, Telnet.

UNIT-II:
Business Models for Business Processes in the Internet of Things, IoT/M2M systems LAYERS AND designs standardizations, Modified OSI Stack for the IoT/M2M Systems, ETSI M2M domains and High- level capabilities, Communication Technologies, Data Enrichment and Consolidation and Device Management Gateway Ease of designing and affordability

UNIT-III:
Design Principles for the Web Connectivity for connected-Devices, Web Communication protocols for Connected Devices, Message Communication protocols for Connected Devices, Web Connectivity for connected-Devices.

UNIT-IV:
Data Acquiring, Organizing and Analytics in IoT/M2M, Applications /Services /Business Processes, IOT/M2M Data Acquiring and Storage, Business Models for Business Processes in the Internet of Things, Organizing Data, Transactions, Business Processes, Integration and Enterprise Systems.

UNIT-V:
Data Collection, Storage and Computing Using a Cloud Platform for IoT/M2M Applications/Services, Data Collection, Storage and Computing Using cloud platform Everything as a service and Cloud Service Models, IOT cloud-based services using the Xively (Pachube/COSM), Nimbits and other platforms Sensor, Participatory Sensing, Actuator, Radio

Frequency Identification, and Wireless, Sensor Network Technology, Sensors Technology, Sensing the World.

Text Books:

1. Internet of Things: Architecture, Design Principles and Applications, Rajkamal, McGraw Hill Higher Education
2. Internet of Things, A.Bahgya and V.Madisetti, University Press, 2015

Reference Books:

1. Designing the Internet of Things, Adrian McEwen and Hakim Cassimally,WileyGetting Started with the Internet of Things, Cuno Pfister, Oreilly

III Year I Semester	DEEP LEARNING LAB	L	T	P	C
		0	0	3	1.5

Course Outcomes: On completion of this course, the student will be able to

- Implement deep neural networks to solve real world problems
- Choose appropriate pre-trained model to solve real time problem
- Interpret the result soft different deep learning models

Software Packages required:
- Keras
- Tensorflow
- PyTorch

List of Experiments:

1. Implement multi-layer perceptron algorithm for MNISTH and written Digit Classification.
2. Design neural network for classifying movie reviews(BinaryClassification) using IMDB dataset.
3. Design a neural Network for classifying news wires (Multiclassclassification) using Reuters dataset.
4. Design neural network for predicting house prices using Boston Housing Price dataset.
5. Build a Convolution Neural Network for MNISTH and written Digit Classification.
6. Build a Convolution Neural Network for simple image (dogs and Cats) Classification
7. Use a pre-trained convolution neural network (VGG16)for image classification.
8. Implement one hot encoding of words or characters.
9. Implement word embeddings for IMDB dataset.
10. Implement a Recurrent Neural Network for IMDB movie review classification problem.

Text Books:

1. RezaZadehandBharathRamsundar,"TensorflowforDeepLearning", O'Reilly publishers, 2018

References:

1. https://github.com/fchollet/deep-learning-with-python-notebooks

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

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

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).

d.

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#
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 Wedges and customize widgets for specific UI elements, Themes
- Understand to include animation apart from fetching data

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 Book:
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 1stEdition, Apres

B. Tech III Year - II Semester

III Year II Semester	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, 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, ISO9000. SEI Capability maturity model. Few other important quality standards, and Six Sigma**.**

UNITV:

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,5thEdition, PHI.
2. Software Engineering: A Practitioner's Approach, Roger S. Pressman, 9thEdition, McGraw Hill International Edition.

Reference Books:

1. Software Engineering, Ian Sommerville,10thEdition, Pearson.
2. Software Engineering, Principles and Practices, DeepakJain, 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_01260589506387148827_shared/overview
3) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013382690411003904735_shared/overview

III Year II Semester	GENERATIVE AI	L	T	P	C
		3	0	0	3

Course Objectives:
• Understand the basics of Generative AI.
• Know the basics of Text Generation.
• Understand the process of generating videos.
• Know about GAN and its variants.

UNIT-I:
Introduction To Gen Ai: Historical Overview of Generative modelling, 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, VariationalAutoencoders, 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, GPT4All, 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.

III Year II Semester	DATA VISUALIZATION	L	T	P	C
		3	0	0	3

Pre-Requisites: Computer Graphics, Image Processing
Course Objective:
- familiarize students with the basic and advanced techniques of information visualization and scientific visualization
- learn key techniques of the visualization process
- a detailed view of visual perception, the visualized data and the actual visualization, interaction and distorting techniques

SYLLABUS:
UNIT-I:
Introduction: What Is Visualization? History of Visualization, Relationship between Visualization and Other Fields. The Visualization Process, Introduction of visual perception, visual representation of data, Gestalt principles, information overloads.

UNIT-II:
Creating visual representations, visualization reference model, visual mapping, visual analytics, Design of visualization applications

UNIT-III:
Classification of visualization systems, Interaction and visualization techniques misleading, Visualization of one, two and multi-dimensional data, text and text documents.

UNIT-IV:
Visualization of groups, trees, graphs, clusters, networks, software, Metaphorical visualization

UNIT-V:
Visualization of volumetric data, vector fields, processes and simulations, Visualization of maps, geographic information, GIS systems, collaborative visualizations, Evaluating visualizations
Recent trends in various perception techniques, various visualization techniques, data structures used in data visualization.

Textbook:
1. WARD, GRINSTEIN, KEIM.Interactive Data Visualization: Foundations, Techniques, and Applications. Natick: A K Peters, Ltd.
2. E. Tufte, The Visual Display of Quantitative Information, Graphics Press.

Resources:
1. https://kdd.cs.ksu.edu/Courses/CIS536/Lectures/Slides/Lecture-34-Main_6up.pdf

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 - BarisBeizer, 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	**CRYPTOGRAPHY & NETWORK SECURITY** (Common to AI&DS, CSE (AI&DS), AI&ML, CSE (AI&ML), CSD, CSE(DS), CSE (AI))	L	T	P	C
		3	0	0	3

Course Objectives:
- Explain the objectives of information security
- Explain the importance and application of each of confidentiality, integrity, authentication and availability
- Understand the basic categories of threats to computers and networks
- Discusses the Mathematics of Cryptography
- Discuss the fundamental ideas of Symmetric and Asymmetric cryptographic Algorithms
- Discusses the Network layer, Transport Layer and Application layer Protocols Enhanced security mechanisms

SYLLABUS:
UNIT – I:
Security Concepts: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security Cryptography.Classical Encryption Techniques-symmetric cipher model, Substitution techniques, Transposition techniques, Rotor Machines, Steganography.

UNIT – II:
Introduction to Symmetric Cryptography: Algebraic Structures-Groups, Rings, Fields, GF(2^n) fields, Polynomials.**Mathematics of Asymmetric cryptography:** Primes, Checking for Primness, Eulers phi-functions, Fermat's Little Theorem, Euler's Theorem, Generating Primes, Primality Testing, Factorization, Chinese Remainder Theorem, Quadratic Congruence, Exponentiation and Logarithm.

UNIT – III:
Symmetric key Ciphers: Block Cipher principles, DES, AES, Blowfish, IDEA, Block cipher operation, Stream ciphers: RC4, RC5
Asymmetric key Ciphers: Principles of public key cryptosystems, RSA algorithm, Diffie-Hellman Key Exchange, Elgamal Cryptographic system, Elliptic Curve Arithmetic, Elliptic Curve Cryptography.

UNIT – IV:
Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Requirements and Security, Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithms (SHA)
Message Authentication Codes: Message Authentication Requirements, Message Authentication Functions, Requirements for Message Authentication Codes, Security of MAC'S, MAC'S Based on Hash Functions: HMAC, MAC'S Based on Block Ciphers: DAA And CMAC
Digital Signatures: Digital Signatures, Elgamal Digital Signature Scheme, Elliptic Curve Digital Signature Algorithm, RSA-PSS Digital Signature Algorithm.

UNIT – V:
Network and Internet Security: Transport-Level Security: Web Security Considerations, Transport Level Security, HTTPS, SSH.
IP Security: IP Security Overview, IP Security Policy, Encapsulating Security Payload, Authentication Header Protocol.
Electronic-Mail Security: Internet-mail Security, Email Format, Email Threats and Comprehensive Email Security, S/MIME, PGP.

Text Books:
1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 7th Edition, 2017
2. Cryptography and Network Security: Behrouz A. Forouzan Debdeep, Mc Graw Hill, 3rd Edition, 2015

Reference Books:
1. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition
2. Introduction to Cryptography with Coding Theory: Wade Trappe, Lawrence C. Washington, Pearson.
3. Modern Cryptography: Theory and Practice byWenbo Mao. Pearson

III Year II Semester	RECOMMENDER SYSTEMS	L	T	P	C
		3	0	0	3

Course Objectives:
This course covers the basic concepts of recommender systems, including personalization algorithms, evaluation tools, and user experiences

Course Outcomes:
- Describe basic concepts behind recommender systems
- Explain a variety of approaches for building recommender systems
- Describe system evaluation methods from both algorithmic and users' perspectives
- Describe applications of recommender systems in various domains

UNIT-I:
Introduction: Recommender system functions, Linear Algebra notation: Matrix addition, Multiplication, transposition, and inverses, covariance matrices, Understanding ratings, Applications of recommendation systems, Issues with recommender system.

UNIT-II:
Collaborative Filtering: User-based nearest neighbor recommendation, Item-based nearest neighbor recommendation, Model based and pre-processing based approaches, Attacks on collaborative recommender systems.

UNIT-III:
Content-based recommendation: High level architecture of content-based systems, Advantages and drawbacks of content-basedfiltering, Item profiles, discovering features of documents, obtaining item features from tags, representing item profiles, Methods for learning user profiles, Similarity based retrieval, Classification algorithms.
Knowledge based recommendation: Knowledge representation and reasoning, Constraint based recommenders, Case based recommenders.

UNIT-IV:
Hybrid approaches: Opportunities for hybridization, Monolithic hybridization design: Feature combination, Feature augmentation, Parallelized hybridization design: Weighted, Switching, Mixed, Pipelined hybridization design: Cascade Meta-level, Limitations of hybridization strategies.

UNIT-V:

Evaluating Recommender System: Introduction, General properties of evaluation research, Evaluation designs, Evaluation on historical datasets, Error metrics, Decision-Support metrics, User-Centered metrics.

Recommender Systems and communities: Communities, collaboration and recommender systems in personalized web search, Social tagging recommender systems, Trust and recommendations

Text Books:

1. Jannach D., Zanker M. and FelFering A., Recommender Systems: An Introduction, Cambridge University Press(2011), 1st ed.
2. Ricci F., Rokach L., Shapira D., Kantor B.P., Recommender Systems Handbook, Springer(2011), 1st ed.

References:

1. Manouselis N., Drachsler H., Verbert K., Duval E., Recommender Systems For Learning, Springer (2013), 1st ed.

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	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 B. Tech II Semester	**COMPUTER VISION**	L	T	P	C
		3	0	0	3

Course Objectives:
- To understand the Fundamental Concepts related to sources, shadows and shading
- To understand the Geometry of Multiple Views

UNIT–I: CAMERAS: Pinhole Cameras Radiometry –Measuring Light: Lightin Space, Light Surfaces, Important Special Cases Sources, Shadows, And Shading: Qualitative Radiometry, Sources and Their Effects, Local Shading Models, Application: Photometric Stereo, **inter reflections**: 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 Get stalt, **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 and 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 and R. E. Woods "Digital Image Processing" Addison Wesley 2008. 3. Richard Szeliski "Computer Vision: Algorithms and Applications" Springer-Verlag London Limited 2011.

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.

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	GENERATIVE AI LAB	L	T	P	C
		0	0	3	1.5

Course Objectives:

1. To learn Python and Tensor Flow skills for Generative AI.

2. To study techniques for cleaning and preparing data for Generative AI tasks.

3. To implement generative AI models

4. To develop innovative applications using generative AI tools and techniques.

Course Outcomes:

After learning the course, students will be able to:

1. Implement Python and Tensor Flow basics, including data handling and preprocessing techniques.

2. Implement Generative AI models such as GANs, VAEs, LSTM networks, and Transformer models for image text, and music generation tasks.

3. Evaluate model performance and experiment with hyperparameters and optimization techniques to enhance Generative AI outcomes.

4. Develop innovative applications in image, text, and music generation, showcasing practical skills

List of Experiments:

1. Write Python scripts to implement basic operations and TensorFlow 2 tensors
2. Implement a Generative Adversarial Network (GAN) architecture using TensorFlow 2. Train theGAN model on a dataset such as MNIST or CIFAR-10 for image generation tasks.
3. Train a GAN model on a custom dataset for image generation. Experiment with hyperparameters, loss functions, and optimization techniques to optimize GAN training.
4. Explore advanced techniques such as Wasserstein GANs, Progressive GANs, or StyleGANs for image generation. Implement and compare these techniques for generating high-quality images.
5. Develop applications for image and video generation using trained Generative AI models. Use the models to generate art, create deep fakes, or synthesize video content.

6. **Text Generation:** Implement a Long Short-Term Memory (LSTM) network using TensorFlow 2 fortext generation tasks. Train the LSTM model on a dataset of text sequences and generate new textsamples.

7. **Text generation:** Implement a Transformer-based language model (e.g., GPT) using TensorFlow 2for text generation. Fine-tune the model on a text corpus and generate coherent and contextuallyrelevant text.

8. **Text generation:** Fine-tune a pre-trained language model (e.g., GPT, BERT) using transfer learningtechniques. Fine-tune the model on a domain-specific dataset and evaluate its performance for textgeneration tasks.

9. **Text generation:** Develop applications for text generation tasks such as story generation, dialoguegeneration, or code generation using trained Generative AI models.

10. **Music Generation:**Preprocess music data and represent it in a suitable format for music generation tasks. Explore MIDIor audio representations for training Generative AI models.

11. **Music Generation:**Implement a Long Short-Term Memory (LSTM) network using TensorFlow 2 for music generation.Train the LSTM model on a dataset of music sequences and generate new musical compositions.

12. **Generate Novel Music Compositions:**Transformer-based Music Generation: Implement a Transformer-based architecture (e.g.,MusicBERT, MusicGPT) using TensorFlow 2 for music generation. Fine-tune the model on a musicdataset and generate novel music compositions.

References:

1. Responsible AI: Implementing Ethical and Unbiased Algorithms, by Shashin Mishra and Sray Agarwal

2. Generative AI in Practice: 100+ Amazing Ways Generative Artificial Intelligence is Changing Business andSociety, Bernard Marr

3. "Generative AI with Python and TensorFlow 2: Create images, text, and music with VAEs, GANs, LSTMs,Transformer models", Joseph Babcock and Raghav Bali

4. "Generative Adversarial Networks: An Overview" by Vinod Nair and Geoffrey E. Hinton.

5. "Hands-On Generative Adversarial Networks with PyTorch 1.x" by Stefano Bosisio and Vijayabhaskar J.

III Year – II Semester	DATA VISUALIZATION LAB	L	T	P	C
		0	0	3	1.5

Course Objectives:

- To visualize the different datasets using histograms, line charts.
- To understand the use of bar charts and box plots.
- To understand Scatter plots, mosaic plots
- To understand different Map visualizations
- To learn advanced graphs such as correlogram, heatmap and 3D graphs.

Course Outcomes: At the end of the course student will be able to

- Visualize the different datasets using histograms, line charts.
- Makeuse of bar charts and box plots on different datasets
- Apply Scatter plots, mosaic plots in R for different datasets
- Apply different Map visualizations in R
- Create advanced graphs such as correlogram, heatmap and 3D graphs.

List of Experiments:

1. a) Load VADeaths(Death Rates in Virginia)dataset in R and visualize the data using different histograms.

    b) Load air quality dataset in R and visualize La Guardia Airport's dialy maximum temperature using histogram.
2. Load AirPassengers dataset in R and visualize the data using line chart that shows increase in air passengers over given time period.
3. a) Load iris dataset in R, visualize the data using different Bar Charts and also demonstrate the use of stacked plots.

    b)Load air quality dataset in R and visualize ozone concentration in air.
4. a) Load iris dataset in R, visualize the data using different Box plots including group by option and also use color palette to represent species.

    b) Load air quality dataset in R and visualize air quality parameters using box plots.
5. Visualize iris dataset using simple scatter, multivariate scatter plot and also visualize scatter plot matrix to visualize multiple variables across each other.
6. Load diamonds dataset in R and visualize the structure in datasets with large data points using hexagon binning and also add color palette then use the
7. Load HairEyeColor dataset in R and plot categorical data using mosaic plot.
8. Load mtcars dataset in R and visualize data using heat map.
9. Install leaflet library in R and perform different map visualizations.
10. Visualize iris dataset using 3d graphs such as scatter3d, cloud, xyplot.
11. Make use of correlogram to visualize data in correlation matrices for iris dataset.
12. Install maps library in R and draw different map visualizations.

Web References:

1. https://www.analyticsvidhya.com/blog/2015/07/guide-data-visualization-r/
2. https://www.geeksforgeeks.org/data-visualization-in-r/

III Year II Semester	SOFT SKILLS	L	T	P	C
	(Skill Enhancement Course)	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

	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	**L**	**T**	**P**	**C**
	MANAGEMENT SYSTEMS	**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_012758066672820224 56_shared/overview

	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

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_01260589506387148827_shared/overview
3) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013382690411003904735_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

Syllabus

for

Minors

	DATA BASE 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 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

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, Ramez Elmasri, 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_01275806667282 02245 6_shared/overview

		L	T	P	C
DATA BASE 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,
- 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

	ARTIFICIAL INTELLIGENCE	L	T	P	C
		3	0	0	3

Course Objectives:
- Gain a historical perspective of Artificial Intelligence (AI) and its foundations.
- Become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
- Investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- Experience AI development tools such as an 'AI language', expert system shell, and/or data mining tool. Experiment with a machine learning model for simulation and analysis.
- Explore the current scope, potential, limitations, and implications of intelligent systems.

Course Outcomes (COs): At the end of the course, student will be able to
- Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents.
- Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game-based techniques to solve them.
- Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing.
- Attain the capability to represent various real life problem domains using logic-based techniques and use this to perform inference or planning.
- Solve problems with uncertain information using Bayesian approaches.

UNIT-I:
Introduction to artificial intelligence: Introduction, history, intelligent systems, foundations of AI, applications, tic-tac-tie game playing, development of AI languages, current trends in AI, **Problem solving: state-space search and control strategies:** Introduction, general problem solving, characteristics of problem, exhaustive searches, heuristic search techniques, iterative-deepening a*, constraint satisfaction

UNIT-II:
Problem reduction and game playing: Introduction, problem reduction, game playing, alpha-beta pruning, two-player perfect information games, **Logic concepts:** Introduction, propositional calculus, proportional logic, natural deduction system, axiomatic system, semantic tableau system in proportional logic, resolution refutation in proportional logic, predicate logic

UNIT-III:
Knowledge representation: Introduction, approaches to knowledge representation, knowledge representation using semantic network, extended semantic networks for KR, knowledge representation using frames, **advanced knowledge representation techniques:** Introduction, conceptual dependency theory, script structure, cyc theory, case grammars, semantic web.

UNIT-IV:
Uncertainty measure: probability theory: Introduction, probability theory, Bayesian belief networks, certainty factor theory, dempster-shafer theory

UNIT-V:

Fuzzy sets and fuzzy logic: Introduction, fuzzy sets, fuzzy set operations, types of membership functions, multi valued logic, fuzzy logic, linguistic variables and hedges, fuzzy propositions, inference rules for fuzzy propositions, fuzzy systems.

Text Books:
1. Artificial intelligence, A modern Approach, 2nded, Stuart Russel, Peter Norvig, Prentice Hall
2. Artificial Intelligence, Saroj Kaushik, 1st Edition, CENGAGE Learning, 2011.

Reference Books:
1. Artificial intelligence, structures and Strategies for Complex problem solving, 5th Edition, George F Lugar, PEA
2. Introduction to Artificial Intelligence, Ertel, Wolf Gang, Springer, 2017
3. Artificial Intelligence, A new Synthesis, 1st Edition, Nils J Nilsson, Elsevier, 1998
4. Artificial Intelligence- 3rd Edition, Rich, Kevin Knight, Shiv Shankar B Nair, TMH
5. Introduction To Artificial Intelligence and Expert Systems, 1st Edition, Patterson, Pearson India, 2015

	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

	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 dead lock 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: Multi threading 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, Dead lock characterization, Methods for handling Dead locks, Dead lock 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, GalvinPB, GagneG, 10thEdition, Wiley, 2018.
2. Modern Operating Systems, TanenbaumAS,4thEdition, Pearson ,2016.

Reference Books:

1. Operating Systems -Internals and Design Principles, Stallings W, 9thedition, Pearson, 2018
2. Operating Systems: A Concept Based Approach, D.MD hamdhere, 3rdEdition, McGraw- Hill, 2013

Online Learning Resources:

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

B. Tech – IV Year I Semester

S.No	Category	Title	L	T	P	C
1	Professional Core	Natural Language Processing	3	0	0	3
2	Management Course- II	Human Resources and Project Management	2	0	0	2
3	Professional Elective-IV	1. Software Architecture & Design Patterns 2. Blockchain Technology 3. Quantum Computing 4. Augmented Reality and Virtual Reality 5. Any of the 12-Week SWAYAM /NPTEL Course suggested by the BoS	3	0	0	3
4	Professional Elective-V	1. Agile Methodologies 2. Architecture for Management of Large Datasets 3. Reinforcement Learning 4. High Performance Computing 5. Any of the 12-Week SWAYAM /NPTEL Course suggested 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	**19**	**1**	**2**	**21**
MC	Student may select from the Same minor pool		3	0	0	3
HC	Student may select from the Same honors pool		3	0	0	3
HC	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	Credits
1	Internship / Project Work	Full semester Internship & Project Work	0	0	24	12

Note: Student need to do at least ONE MOOC Course (3 credits out of 160 credits) to meet the mandatory requirement (11ᵗʰ criteria, as per R23 Regulations)

Open Electives, offered to other department students:

Open Elective I: Operating Systems / Computer Organization and Architecture

Open Elective II: Database Management Systems

Open Elective III: Object Oriented Programming Through Java / Quantum Science and Technology

Open Elective IV: Computer Networks / Software Engineering / Data Structures for Data Science

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*

For Minor in AI: **L-T-P-C**

1. Database Management Systems   3-0-3-4.5 (III-I)
2. Introduction to Artificial Intelligence   3-0-0-3 (III-I)
3. Advanced Data Structures & Algorithm Analysis   3-0-3-4.5 (III-II)
4. Mathematics for Machine Learning*   3-0-0-3 (IV-I)
   Any of the following 12 Week 3 credit NPTEL MOOC Courses
5. Object Oriented Programming through Java
6. Computer Networks and Internet Protocol
7. Advanced Data Structures & Algorithm Analysis
8. Operating Systems
9. Software Engineering
10. Deep Learning

Note: As per AICTE APH 2023-24 student can opt any one of the following streams:

1. Honors in CSE
2. Honors in AI&ML
3. Honors in Data Science

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	**NATURAL LANGUAGE PROCESSING** (Professional Core)	**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.

Course Outcomes: After completion of this course
- Demonstrate a given text with basic Language features
- To design an innovative application using NLP components
- Explain a rule based system to tackle morphology/syntax of a language
- To design a tag set to be used for statistical processing for real-time applications
- To compare and contrast the use of different statistical approaches for different types of NLP applications.

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, Word embedding's and associated techniques.

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, 2nd Edition, 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, 1st Edition, Breck Baldwin, Atlantic Publisher, 2015.
2. Natural Language Processing with Java, 2nd Edition, Richard M Reese, OReilly Media,2015.
3. Handbook of Natural Language Processing, Second, Nitin Indurkhya and Fred J. Damerau, Chapman and Hall/CRC Press, 2010.Edition
4. Natural Language Processing and Information Retrieval, 3rd Edition, Tanveer Siddiqui, U.S. Tiwary, Oxford University Press,2008.

IV Year I Semester	HUMAN RESOURCES AND PROJECT MANAGEMENT (Management Course-II)	L	T	P	C
		3	0	0	3

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 Outcomes
- 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 of 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 Outcomes: At the end of the Course student will be able to:
CO1: Discuss the Cryptographic primitives used in Blockchain (K2)
CO2: Discuss about various technologies borrowed in blockchain (K2)
CO3: Illustrate various models for blockchain (K2)
CO4: Discuss about Ethereum (K2)
CO5: Discuss about Hyperledger Fabric (K2)

UNIT I INTRODUCTION TO BLOCKCHAIN: Introduction, history of Bitcoin and origins of Blockchain, Fundamentals of Blockchain and key components (Chapter 1-book1), Permission and Permission-less platforms(Chapter 1-book2), Introduction to Cryptography, SHA256 and ECDSA, Hashing and Encryption, Symmetric/ Asymmetric keys, Private and Public Keys(Chapter 3-book2).

UNIT II   TECHNOLOGIES BORROWED IN BLOCKCHAIN: Technologies Borrowed in Blockchain –hash pointers- - Digital cash etc.- Bitcoin blockchain - Wallet – Blocks Merkley Tree - hardness of mining - Transaction verifiability - Anonymity - forks - Double spending - Mathematical analysis of properties of Bitcoin - Bitcoin- the challenges and solutions. (Chapter 3-book2).

UNIT III CONSENSUS MECHANISMS: Consensus Algorithms: Proof of Work (PoW) as random oracle - Formal treatment of consistency- Liveness and Fairness - Proof of Stake (PoS) based Chains -Hybrid models (PoW + PoS), Byzantine   Models of fault tolerance. ((Chapter 1-book2))

UNIT IV   ETHEREUM: Ethereum -Ethereum Virtual Machine (EVM) -Wallets for Ethereum -Solidity - Smart Contracts **(Chapter 5-book1),** - The Turing Completeness of Smart Contract Languages and verification challenges- Using smart contracts to enforce legal contracts- Comparing Bitcoin scripting vs. Ethereum Smart Contracts-Some attacks on smart contracts **(Chapter 6 and Chapter 7-book2)**

UNIT V   HYPERLEDGER FABRIC   : Hyperledger fabric-   the   plug   and   play platform and mechanisms in permissioned blockchain - Beyond Cryptocurrency – applications of blockchain in cyber security- integrity of information- E-Governance and other contract enforcement mechanisms - Limitations of blockchain as a technology   and myths vs reality of blockchain technology **(Chapter 16-book1),** (**Chapter 9 -book2)**

TEXT BOOKS:
1. Blockchain Technology Chandramouli Subramanian, Asha A George, Abhilash K A and Meena Karthikeyan, University Press, 2020.
2. Mastering Blockchain - Distributed ledger technology, decentralization, and smart contracts explained, Imran Bashir,2nd ed. Edition,2018, pakct publication

REFERENCES:

1. S.Shukla, M.Dhawan, S.Sharma,S. Venkatesan "Blockchain Technology: Cryptocurrency and Applications" ,Oxford University Press 2019 .
2. Cryptography and network security principles and practice, William Stallings, Pearson, 8th edition,

WEB REFERENCES:

1. https://drive.google.com/file/d/1PtYaDmWYaqPVGjKDnMYGWO5eoI5wMPtJ/view
2. https://archive.nptel.ac.in/courses/106/104/106104220/
3. https://www.tutorialspoint.com/blockchain/index.htm

IV Year I Semester	QUANTUM COMPUTING (Professional Elective-IV)	L	T	P	C
		3	0	0	3

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

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. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge

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

IV Year I Semester	AUGMENTED REALITY & VIRTUAL REALITY (Professional Elective-IV)	L	T	P	C
		3	0	0	3

Course Objectives:
- Provide a foundation to the fast growing field of AR and make the students aware of the various AR concepts.
- To give historical and modern overviews and perspectives on virtual reality. It describes the fundamentals of sensation, perception, technical and engineering aspects of virtual reality systems.

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. Allan Fowler-AR Game Development‖, 1st Edition, A press 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 Published by O'Reilly Media, Inc., 2016,ISBN:9781491962381
5. Sanni Siltanen- Theory and applications of marker-based augmented reality. Julkaisija – Utgivare Publisher. 2012. ISBN 978-951-38-7449-0
6. Gerard Jounghyun Kim, "Designing Virtual Systems: The Structured Approach", 2005

IV Year I Semester	AGILE METHODOLOGIES (Professional Elective-V)	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course the student will be able to:

CO1 Apply agile methodology and agile process to create high quality software.

CO2 Use Agile methodology for knowledge management.

CO3 Apply Agile development and testing techniques to manage risks.

CO4 Analyze the pros and cons of working in Agile Team.

CO5 Apply Feature Driven Development on large size projects.

UNIT –I: Agile Methodology: Theories for Agile Management, Agile Software Development – Traditional Model vs. Agile Model, Classification of Agile Methods, Agile Manifesto and Principles, Agile Project Management, Agile Team Interactions, Ethics in Agile Teams, Agility in Design, Testing, Agile Documentations, Agile Drivers, Capabilities and Values.

UNIT-II: Agile Process: Lean Production - SCRUM, Crystal, Feature Driven Development-Adaptive Software Development - Extreme Programming: Method Overview – Lifecycle – Work Products, Roles and Practices.

UNIT-III: Agile Knowledge Sharing – Role of Story-Cards – Story-Card Maturity Model (SMM).

UNIT-IV: Agility and Requirements Engineering: Impact of Agile Processes in RE, Current Agile Practices, Variance, Overview of RE Using Agile, Managing Unstable Requirements, Requirements Elicitation, Agile Requirements Abstraction Model, Requirements Management in Agile Environment, Agile Requirements Prioritization, Agile Requirements Modelling, Generation – Concurrency in Agile Requirements Generation.

UNIT-V: Agility and Quality Assurance: Agile Product Development, Agile Metrics, Feature Driven Development (FDD), Financial and Production Metrics in FDD, Agile Approach to Quality Assurance, Test Driven Development, Agile Approach in Global Software Development.

TEXT BOOKS:

1. David J. Anderson and Eli Schragenheim, Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results, Pearson, 2003.
2. Hazza and Dubinsky, Agile Software Engineering, Springer, 2009th edition, 2008.

REFERENCE BOOKS:

1. Craig Larman, ―Agile and Iterative Development, Addison-Wesley, First Edition, 2003.

IV Year I Semester	ARCHITECTURE FOR MANAGEMENT OF LARGE DATASETS (Professional Elective-V)	L	T	P	C
		3	0	0	3

Course Objective:
- The objective of this course is to introduce students to the foundational and advanced concepts of big data processing frameworks such as Hadoop, Hive, Spark, Storm, and Giraph, and to equip them with the skills required to set up and administer big data environments.

Course Outcomes:
Upon successful completion of the course, students will be able to:
- Understand the Need and Architecture of Hadoop Ecosystem
- Explain Key Big Data Terminologies and Administer Hadoop
- Work with Hive and Understand Giraph Architecture
- Develop Applications Using Spark and Understand Storm Architecture
- Classify Digital Data and Understand Large Language Models

Unit 1. Hadoop, HDFS and Setting up a Hadoop Cluster
Why Hadoop, Why not RDBMS, RDMBS vs Hadoop, HDFS Daemons, Anatomy of File Read, Anatomy of File Write, Replica Placement Strategy, Working with HDFS commands , Setting up a Hadoop Cluster-Cluster Specification, Cluster Setup and Installation, Hadoop Configuration, Security

Unit 2. Terminologies and Administering Hadoop
Terminologies used in Big Data Environments - In-memory analytics, In-database processing, Symmetric Multiprocessor System(SMP), Massively Parallel Processing, Difference between Parallel and Distributed Systems, Shared Nothing Architecture, Administering Hadoop-Administering HDFS, Monitoring and Maintenance

Unit 3. Hive and Architecture of Giraph
What is Hive, Hive Architecture, Hive Data Types, Hive File Format, HiveQL, Tables, Querying Data, User Defined Functions (UDF), Architecture of Giraph

Unit 4. Spark and Architecture of Storm
Installing Spark, Spark Applications, Jobs, Stages, Tasks, Resilient Distributed Datasets, Shared Variables, Anatomy of a Spark Job Run, Executors and Cluster managers, Architecture of Storm

Unit 5. Classification of Digital Data and Large Language Models for Graphs
Classification of Digital Data - Structured Data, Semi-structured Data and Unstructured Data, Understanding large language models-What is an LLM, Applications of LLMs, Stages of building and using LLMs, Introducing the transformer architecture, Utilizing large datasets, A closer look at the GPT architecture, Building a large language model.

Text Books:

1. Seema Acharya and Suhashini Chellappan, "Big Data Analytics", 2nd Ed, Wiley, 2015
2. Tom White, "Hadoop: The Definitive Guide", 4th Ed, O'reilly, 2015
3. V. K. Jain, Big Data and Hadoop, Khanna Book Publishing, 2020
4. Build a Large Language Model (From Scratch), Sebastian Raschka, September 2024 ISBN 9781633437166

Reference Books:

1. Michael Minelli, Michelle Chambers and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013
2. P. J. Sadalage, M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012
3. Storm https://cds.iisc.ac.in/wp-content/uploads/DS256.2017.Storm_.Tutorial.pdf
4. Giraph https://cds.iisc.ac.in/wp-content/uploads/DS256.2017.L12.Pregel.pdf

IV Year I Semester	REINFORCEMENT LEARNING (Professional Elective-V)	L	T	P	C
		3	0	0	3

COURSE OBJECTIVE:
To provide the fundamentals of Reinforcement learning.

COURSE OUTCOMES:
- Enumerate the elements of Reinforcement Learning
- Solve then-armed Bandit problem
- Compare different Finite Markov Decision Process
- Discuss about Monte Carlo Methods in solving real world problems
- List the Applications and Case Studies of Reinforcement Learning

UNIT-I: The Reinforcement Learning Problem: Reinforcement Learning, Examples, Elements of Reinforcement Learning, Limitations and Scope, An Extended Example: Tic-Tac Toe, Summary, History of Reinforcement Learning.

UNIT-II: Multi-arm Bandits: An n-Armed Bandit Problem, Action-Value Methods, Incremental Implementation, Tracking a Nonstationary Problem, Optimistic Initial Values, Upper-Confidence-Bound Action Selection, Gradient Bandits, Associative Search (Contextual Bandits)

UNIT-III: Finite Markov Decision Processes: The Agent–Environment Interface, Goals and Rewards, Returns, Unified Notation for Episodic and Continuing Tasks, The Markov Property, Markov Decision Processes, Value Functions, Optimal Value Functions, Optimality and Approximation.

UNIT-IV: Monte Carlo Methods: Monte Carlo Prediction, Monte Carlo Estimation of Action Values, Monte Carlo Control, Monte Carlo Control without Exploring Starts, Off- policy Prediction via Importance Sampling, Incremental Implementation, Off-Policy Monte Carlo Control, Importance Sampling on Truncated Returns

UNIT-V: Applications and Case Studies:TD-Gammon, Samuel's Checkers Player, The Acrobot, Elevator Dispatching, Dynamic Channel Allocation, Job-Shop Scheduling.

TEXT BOOKS:
1. Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning-An Introduction",2ndEdition, The MIT Press,2018

2. Marco Wiering, Martijn van Otterlo Reinforcement Learning: State-of-the Art(Adaptation,Learning,andOptimization(12))2012thEdition

REFERENCE BOOKS:
1. Vincent François-Lavet,Peter Henderson,Riashat Islam, An Introduction to Deep Reinforcement Learning (Foundations and Trends(r) in Machine Learning) , 2019

IV Year I Semester	**HIGH PERFORMANCE COMPUTING** **(Professional Elective-V)**	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of the course is to study parallel computing hardware and programming models, performance analysis and modeling of parallel programs

Course Outcomes:

On completion of the course, student will be able to–
- Describe different parallel architectures, inter-connect networks, programming models
- Develop an efficient parallel algorithm to solve given problem
- Analyze and measure performance of modern parallel computing systems
- Build the logic to parallelize the programming task

Unit I Introduction: Motivating Parallelism, Scope of Parallel Computing, Parallel Programming Platforms: Implicit Parallelism, Trends in Microprocessor and Architectures, Limitations of Memory, System Performance, Dichotomy of Parallel Computing Platforms, Physical Organization of Parallel Platforms, Communication Costs in Parallel Machines, Scalable design principles, Architectures: N-wide superscalar architectures, Multi-core architecture.

Unit II Parallel Programming: Principles of Parallel Algorithm Design: Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models, The Age of Parallel Processing, the Rise of GPU Computing, A Brief History of GPUs, Early GPU.

Unit III Basic Communication: Operations- One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction, All-Reduce and Prefix-Sum Operations, Scatter and Gather, All-to-All Personalized Communication, Circular Shift, Improving the Speed of Some Communication Operations. Programming shared address space platforms: threads- basics, synchronization, OpenMP programming

Unit IV: Analytical Models: Sources of overhead in Parallel Programs, Performance Metrics for Parallel Systems, and The effect of Granularity on Performance, Scalability of Parallel Systems, Minimum execution time and minimum cost, optimal execution time. Dense Matrix Algorithms: MatrixVector Multiplication, Matrix-Matrix Multiplication.

Unit V : Parallel Algorithms- Sorting and Graph : Issues in Sorting on Parallel Computers, Bubble Sort and its Variants, Parallelizing Quick sort, All-Pairs Shortest Paths, Algorithm for sparse graph, Parallel Depth-First Search, Parallel BestFirst Search. **CUDA Architecture** : CUDA Architecture, Using the CUDA Architecture, Applications of CUDA Introduction to CUDA C-Write and launch CUDA C kernels, Manage GPU memory, Manage communication and synchronization, Parallel programming in CUDA- C.

Text Books:

1. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, "Introduction to Parallel Computing", 2nd edition, Addison-Wesley, 2003, ISBN: 0-201-64865-2
2. Jason sanders, Edward Kandrot, "CUDA by Example", Addison-Wesley, ISBN-13: 978-0-13-138768-3

Reference Books:

1. Kai Hwang, "Scalable Parallel Computing", McGraw Hill 1998, ISBN:0070317984
2. Shane Cook, "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs", Morgan Kaufmann Publishers Inc. San Francisco, CA, USA 2013 ISBN: 9780124159884
3. David Culler Jaswinder Pal Singh, "Parallel Computer Architecture: A Hardware/ Software Approach", Morgan Kaufmann,1999, ISBN 978-1-55860-343-1
4. Rod Stephens, "Essential Algorithms", Wiley, ISBN: ISBN: 978-1-118-61210-1

	MATHEMATICS FOR MACHINE LEARNING	**L**	**T**	**P**	**C**
IV Year I Semester		**3**	**0**	**0**	**3**

Course Objectives:

- The main objective of the course is to provide a deep understanding of the math that makes machine learning algorithms work.

Course Outcomes:

- At the end of the course the student should be able to
- Understand concepts in abstract algebra and algebraic structures
- Utilize eigen values and matrix factorisation techniques in practical problems
- Comprehend vectors and vector spaces
- Use Probability, Bayes theorem and random variables in applications

UNIT-I: Linear Algebra: Systems of Linear Equations, Matrices, Solving Systems of Linear Equations, Vector Spaces, Linear Independence, Basis and Rank, Linear Mappings, Affine Spaces

UNIT-II: Analytic Geometry: Norms, Inner Products, Lengths and Distances, Angles and Orthogonality, Orthonormal Basis, Orthogonal Complement, Inner Product of Functions, Orthogonal Projections, Rotations

UNIT-III: Matrix Decompositions: Determinant and Trace, Eigen values and Eigenvectors, Cholesky Decomposition, Eigen decomposition and Diagonalization, Singular Value Decomposition, Matrix Approximation, Matrix Phylogeny

UNIT-IV: Vector Calculus : Differentiation of Univariate Functions, Partial Differentiation and Gradients, Gradients of Vector-Valued Functions, Gradients of Matrices, Useful Identities for Computing Gradients, Backpropagation and Automatic Differentiation, Higher-Order Derivatives, Linearization and Multivariate Taylor Series

UNIT-V: Probability and Distributions: Construction of a Probability Space, Discrete and Continuous Probabilities, Sum Rule, Product Rule, and Bayes' Theorem, Summary Statistics and Independence, Gaussian Distribution, Conjugacy and the Exponential Family, Change of Variables/Inverse Transform Continuous Optimization: Optimization Using Gradient Descent, Constrained Optimization and Lagrange Multipliers, Convex Optimization.

Text Books:

1. "Mathematics for Machine Learning", Marc Peter Deisenroth, A. Aldo Faisal and Cheng Soon Ong, Cambridge University Press.
2. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer 2017.

Reference Books:

1. Machine Learning: An Applied Mathematics Introduction, Paul Wilmott, Panda Ohana Publishing 2019.

IV Year I Semester	QUANTUM SCIENCE AND TECHNOLOGY	L	T	P	C
		3	0	0	3

Prerequisites: Basic Physics, Linear Algebra, and Introduction to Modern Physics

Course Objectives:
1. To introduce fundamental concepts of quantum mechanics and its mathematical formalism.
2. To explore quantum computing and communication principles and technologies.
3. To understand the physical implementation and limitations of quantum systems.
4. To enable students to relate quantum theory to practical applications in computing, cryptography, and sensing.
5. To familiarize students with the emerging trends in quantum technologies.

Course Outcomes:
After completing this course, students will be able to:
CO1. Explain core principles of quantum mechanics and their technological implications.
CO2. Analyze quantum phenomena like superposition and entanglement.
CO3. Apply mathematical tools to model and solve quantum systems.
CO4. Demonstrate understanding of quantum algorithms and quantum circuits.
CO5. Evaluate potential applications and challenges in quantum communication and sensing.

Unit 1: Fundamentals of Quantum Mechanics: Historical background: Blackbody radiation, photoelectric effect, and Compton scattering; Dual nature of light and matter; De Broglie hypothesis; Schrödinger equation; Free particle, infinite potential well, step potential; Operators and observables: position, momentum, Hamiltonian; Commutation relations and uncertainty principle; Quantum postulates and measurement theory; Eigenvalues, eigenfunctions.

Unit 2: Quantum Information Theory: Classical vs. quantum information; Qubit representation using Bloch sphere; Quantum superposition and quantum entanglement; Dirac notation (bra-ket), tensor products, and composite systems; Bell states and EPR paradox; Quantum gates: Pauli-X, Y, Z; Hadamard; Phase; T; CNOT; Quantum circuit models and notation; Measurement in computational basis; Quantum teleportation and no-cloning theorem; Quantum state tomography (introductory)

Unit 3: Quantum Computing: Classical computing review and limitations; Quantum parallelism and interference; Deutsch and Deutsch-Jozsa algorithms; Grover's search algorithm, Oracle and amplitude amplification; Shor's factoring algorithm (overview and significance); Quantum Fourier Transform (QFT); Quantum error correction: Bit-flip, phase-flip, and Shor's 9-qubit code; Introduction to quantum programming: Qiskit, Cirq, IBM Quantum Experience (overview)

Unit 4: Quantum Communication: Introduction to quantum cryptography; Quantum key distribution (QKD): BB84 protocol; Entanglement-based QKD: Ekert protocol (E91); Eavesdropping and security of QKD; Quantum teleportation (circuit and protocol); Quantum dense coding; Quantum networks and entanglement swapping; Role of quantum repeaters; Single-photon sources and

detectors; Implementation challenges (loss, decoherence, noise)

Unit 5: Quantum Technologies and Applications: Quantum sensors: magnetometry, gravimetry; Quantum metrology: standard time, atomic clocks; Quantum imaging and lithography; Quantum materials: topological insulators, graphene, quantum dots; NV centers in diamonds for sensing; Hardware platforms: Superconducting qubits, Trapped ions, Photonic quantum processors; Quantum supremacy and NISQ era; Global initiatives: IBM, Google, D-Wave, IonQ, India's NQM; Ethical concerns and future prospects

Text Books:
1. **"Quantum Computation and Quantum Information"** by Michael A. Nielsen and Isaac L. Chuang
2. **"Quantum Mechanics: Concepts and Applications"** by Nouredine Zettili

Online Learning Resources:

1. https://nptel.ac.in/courses/104104082
2. https://nptel.ac.in/courses/115104096
3. https://nptel.ac.in/courses/122106034

DATA STRUCTURES FOR DATA SCIENCE	L	T	P	C
	3	0	0	3

Course Objectives:
- To introduce fundamental and advanced data structures required for data science applications.
- To enable students to analyze, design, and implement appropriate data structures for real-world data problems.
- To bridge core data structure concepts with data science tools and techniques.

Course Outcomes (COs):
After successful completion of the course, students will be able to:

1. CO1: Explain the role of fundamental data structures in data science applications.
2. CO2: Implement linear and non-linear data structures using Python for real-world datasets.
3. CO3: Apply tree, heap, and graph structures to solve data organization and search problems.
4. CO4: Analyze the efficiency of different data structures and select suitable ones for specific data science tasks.
5. CO5: Design data-driven solutions by integrating appropriate data structures for analytics and machine learning workflows.

UNIT–I: Foundations of Data Structures and Python for Data Science: Introduction to Data Science and role of Data Structures, Review of Algorithm Analysis: Time and Space Complexity (Big-O notation), Abstract Data Types (ADT), Arrays and Dynamic Arrays, Strings and String Processing.

Python Data Structures: Lists, Tuples, Sets, Dictionaries; Introduction to NumPy arrays and vectorized operations

UNIT–II: Linear Data Structures and Applications: Linked Lists (Singly, Doubly) and Applications, Stacks and Queues (Simple, Circular, Priority Queue), Deques

Applications in Data Science: Expression evaluation, Data streaming, Undo/Redo operations, Sliding window problems, Implementation using Python collections

UNIT–III: Trees and Hierarchical Data Structures: Trees: Terminology and Representation; Binary Trees and Binary Search Trees (BST); Tree Traversals (DFS, BFS); Balanced Trees (AVL – basic concept); Heaps and Priority Queues

Applications in Data Science: Decision Trees (conceptual link); Heap in Top-K problems; Hierarchical clustering (data representation perspective)

UNIT–IV: Hashing and Graph Data Structures: Hash Tables and Hash Functions, Collision Handling Techniques, Dictionaries and Sets in Python (using hash-based); Graph Representation (Adjacency Matrix & List); Graph Traversal (BFS, DFS); Shortest Path (Dijkstra's algorithm – basic idea)

UNIT–V: Advanced and Specialized Data Structures for Data Science: Trie (Prefix Trees), Disjoint Sets (Union-Find); Sparse Matrices and Efficient Storage; KD-Trees (basic concept); Bloom Filters (introductory concept)

Data Structures for Big Data (brief overview of distributed data storage concepts); Introduction to Pandas DataFrame as a high-level structured data abstraction

Text Book:

1. Data Science from Scratch (2e): First Principles with Python, Joel Grus
2. Data Structures and Algorithms in Python, Roberto Tamassia, Michael H. Goldwasser, and Michael T. Goodrich
3. Job Ready Python: From first script to Production API, Naren Allam

IV Year I Semester	PROMPT ENGINEERING (Skill Enhancement Course)	L	T	P	C
		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.

		L	T	P	C
IV Year I Semester	**CONSTITUTION OF INDIA** **(Audit Course)**				
		2	0	0	--

Course Objectives:

- Understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.
- To 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.
- To 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: ZilaPachayat, Elected officials and their roles, CEO ZilaPachayat: 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.

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