

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY SRINAGAR



SYLLABUS

FOR

B. TECH. INFORMATION TECHNOLOGY

2023 BATCH

Semester III

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (3 rd Semester)	Data Structure	ITT201	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: C Programming

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

COs	Course Outcome	Bloom's Taxonomy Level
CO1	<u>Analyze</u> the concepts of algorithm evaluation and find time and space complexities for searching and sorting algorithms.	4
CO2	<u>Analyze</u> and <u>compare</u> different sorting algorithms, such as selection sort, insertion sort, exchange sort, and merge sort.	5
CO3	<u>Evaluate</u> basic operations on binary trees, binary search trees.	5
CO4	<u>Analyze</u> and <u>demonstrate</u> the representation and traversal techniques of graphs and their applications	4

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	Introduction: Overview of data structure, Basics of Algorithm Analysis including Running Time Calculations. Abstract Data Types: Arrays, Arrays and Pointers, Multidimensional Array, String processing, General Lists and List ADT, List manipulations, Single, double and circular lists. Stacks and Stack ADT, Stack Manipulation, Prefix, infix and postfix expressions, recursion. Queues and Queue ADT, Queue manipulation.	8
Module 2	Sorting: Sorting concept, order, stability, Selection sorts (Straight, Heap), insertion sort (Straight Insertion, Shell sort), Exchange Sort (Bubble, Quicksort), Merge sort (External Sorting) (Natural merge, Balanced merge and Polyphase merge). Searching: List Search, Sequential search, Binary Search, Comparison and Analysis Internal. Hashing: Hash Function, Hashing methods, Collision Resolution Strategies	12

Module 3	<p>Trees: Basic Terminology and Properties of Trees.</p> <p>Binary Trees: Binary Tree Representation, Array Representation (Binary tree) and Dynamic Representation, Tree manipulation algorithms, Expression trees and their usage.</p> <p>Tree Traversal Algorithms: In order, Preorder, and Postorder, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm.</p> <p>Search Trees: Binary search trees, AVL trees, heaps and their implementation, priority queues, BTrees, B* Tree, B+ trees.</p>	12
Module 4	<p>Graphs: Terminology, Disjoint sets representation, union find algorithm.</p> <p>Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi list.</p> <p>Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees (Prims and Kruskal algorithm), Transitive Closure and Shortest Path algorithm (Warshal Algorithm and Dijkstra Algorithm).</p>	10

Books Recommended:

1. Data structures: A pseudocode approach with C++ by Gilberg, R. F., & Forouzan, B. A. Brooks/Cole Publishing Co.
2. Data structures and algorithms. by Aho Alfred, V., Hopcroft John, E., Ullman Jeffrey, D., Aho Alfred, V., Bracht Glenn, H., Hopkin Kenneth, D., ... & Johnson, C. A, Addison-Wesley.

Other Books Recommended:

1. Introduction to algorithms by Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C, MIT press.
2. Fundamentals of computer algorithms by Horowitz, Galgotia publications.
3. Data structures and algorithm analysis in C++ by Weiss M.A., Pearson Education.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (4 th Semester)	Control Systems	EET254	3-0-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: Differential Equations, Laplace Transform.

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

COs	Course Outcome	Bloom's Taxonomy Level
CO1	<u>Examine</u> control systems for real-world problems using concepts of open-loop, closed-loop, and applying standard control strategies.	4
CO2	<u>Develop</u> mathematical models of LTI systems using various methods.	3
CO3	<u>Determine</u> the time response characteristics of first and second-order LTI systems using standard test signals.	5
CO4	<u>Analyze</u> and <u>Estimate</u> the stability of LTI systems using time response and frequency response analysis techniques.	5

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	Introduction to continuous control systems: Control system: Basics, and Need; Classification of systems: LTI, Open-loop, Closed-loop (automatic and manual control); Effects of feedback.	5
Module 2	Mathematical modeling of LTI systems: Basic governing equations; Transfer function models; Simplified models of complex systems using Block diagram reduction and Signal flow graph. Modeling of physical systems: electrical, mechanical, and electro-mechanical; State space representation of dynamic systems.	12
Module 3	Time domain analysis: Standard test signals, First and second-order systems; Time responses of these systems to standard test signals. Transient, Steady-state, and Error analysis.	10
Module 4	Stability studies: System stability, Stability analysis using Routh's criterion, and Root-locus technique. Stability analysis using frequency response Polar plot, and Bode plot.	10
Module 5	Controllers and compensators: Design and tuning of Proportional (P), Integral (I), and Derivative (D) controllers. Lead-lag compensators.	5

Text Books:

1. Control Systems Engineering, Norman S. Nise, Wiley.
2. Control Systems (Principles and Design), M. Gopal, Tata McGraw-Hill.

Reference Books:

1. Control systems, Anand Kumar, PHI Learning Private Limited.
2. Feedback control of dynamic systems, Franklin and Powel, Prentice Hall.
3. Problems and solutions of control systems, A.K. Jairath, CBS Publishers & Distributors Pvt. Ltd.
4. Design of feedback control systems, Stefani, Oxford University Press.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (3 rd Semester)	Logic and Graph Theory	ITT 202	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: None

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

COs	Course Outcome	Bloom's Taxonomy Level
CO1	<u>Apply</u> basic operations of Propositional logic and Boolean Algebra.	3
CO2	<u>Analyze</u> various proof techniques.	4
CO3	<u>Discover</u> the basics of graph theory and implement them for real-world problems.	3
CO4	<u>Review</u> trees, their properties, traversal modes, and types.	2

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	Propositional Calculus: Statements, Basic operations, the Truth value of compound statements, Algebra of Propositions, Tautologies, and contradiction, Conditional and Bi-conditional statements, logical implications, logical equivalence, predicates, Universal and existential quantifiers. Logic gates, Boolean Algebra, Postulates of Boolean Algebra; Theorems of Boolean Algebra, Sum of products and product of sums Simplification.	13
Module 2	Proof techniques: Notions of implication, converse, inverse, contrapositive, negation, and contradiction; the structure of formal proofs; direct proofs; proof by counterexample; proof by contraposition; proof by contradiction; mathematical induction.	7
Module 3	Graph Theory: Graphs and Multi-graphs, graph representation, Degree of a vertex, Paths, connectivity, Cut points, Bridges, Walks, paths, cycles, connected graphs, Bipartite, Regular, Planar and connected graphs, Euler graphs, Euler's theorem, Hamiltonian path and circuits, Graph coloring, chromatic number, isomorphism and Homomorphism of graphs, Konigsberg seven bridge problem, Shortest path problem.	16
Module 4	Trees: Introduction to Trees, types and properties of trees, pendant vertices in trees, degree sequences in trees, necessary and sufficient conditions for a sequence to be a degree sequence of a tree, tree traversals, spanning tree, minimum spanning trees.	6

Recommended Books:

1. Kenneth H. Rosen. Discrete Mathematics and its Applications, Tata McGraw Hill.
2. C. L. Liu. Elements of Discrete Mathematics, 2nd Ed. Tata McGraw Hill.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
B. Tech. Information Technology Engg. 2 nd Year (3 rd Semester)	Discrete Mathematics	MAT-208	3-0-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: A student should have basic knowledge of set theory.

Course Outcomes: At the end of the course, a student should be able to:

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	Apply the concept of sets, relations and functions in various engineering problems.	3
CO2	Solve problems by using counting techniques.	3
CO3	Determine ordered sets and lattices.	5
CO4	Evaluate engineering problems using concepts of groups and rings.	5

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	Sets, Review of Basic Set Operations, the cardinality of a set, countability. Relations, Types of relations, recurrence relations, operations of relations and applications. Functions, types of functions, generating functions, operations of functions and applications.	14
Module 2	Basic counting principles, permutations and combinations, pigeon hole principle with applications, inclusion-exclusion principle, introduction to special numbers.	6
Module 3	Ordered sets, Diagram of Partially ordered sets, Supremum and Infimum, well-ordered sets, Lattices, Bounded and complemented lattice, Distributive Lattice, topological ordering, Congruence arithmetic.	8
Module 4	Groups, semigroup, infinite group, Finite group, order of a group, Abelian group, subgroup, Lagrange's Theorem, Cosets, Normal Subgroups, order of an element of a group, cyclic group, Rings, Homomorphism and Isomorphism of rings.	14

Recommended Books:

1. S. Lipschutz, and M. L. Lipson, *Schaum's outline Series, Discrete Mathematics*, Tata Mc-Graw Hill Publication, (2007).
2. C.L. Liu, *Elements of discrete mathematics*, Tata Mc-Graw Hill Publication, (2017).
3. I. N. Herstein, *Topics in Algebra*, John Wiley & Sons, (1996).
4. Lipschutz, Seymour, and Marc Lars Lipson. *Theory and problems of discrete mathematics*. McGraw-Hill, 2007.
5. K. Bogart, S. Drysdale, C. Stein. *Discrete Math for Computer Science Students*.
6. West, Douglas Brent. *Introduction to graph theory*. Vol. 2. Upper Saddle River: Prentice hall.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (3 rd Semester)	Object Oriented Programming	ITT203	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: C Programming

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

COs	Course Outcome	Bloom's Taxonomy Level
CO1	<u>Evaluate</u> the concepts of object-oriented programming for problem solving and critical analysis with introduction to JAVA programming language	5
CO2	<u>Analyze</u> the concepts of Java programming language in the development of small to medium-sized application programs	4
CO3	<u>Criticize</u> the basic operations, loops and functions of Java	5
CO4	<u>Design</u> the graphical user interfaces, multithreaded programming, and event-driven applications using JAVA.	6

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	<p>UNIT I - INTRODUCTION: Programming language Types and Paradigms, Computer Programming Hierarchy, How Computer Architecture Affects a Language? Why Java? Role of Java Programmer in Industry, Features of Java Language, JVM –The heart of Java , Java's Magic Bytecode. The Java Environment: Installing Java, Java Program Development , Java Source File Structure,Compilation, Executions.</p>	8
Module 2	<p>UNIT II - OBJECT-ORIENTED PROGRAMMING CONCEPTS: Basic Language Elements: Lexical Tokens, Identifiers, Keywords, Literals, Comments, Primitive Data Types, Operators Assignments. Object-Oriented Programming: Class Fundamentals , Object & Object reference , Object Lifetime & Garbage Collection, Creating and Operating Objects , Constructor & initialization code block, Access Control, Modifiers, methods Nested , Abstract Class & Interfaces, Defining Methods, Argument Passing Mechanism, Method Overloading, Recursion, Dealing with Static Members, Finalize() Method, Native Method.</p>	13

Module 3	<p>UNIT III - EXTENDING CLASSES AND INHERITANCE Use and Benefits of Inheritance in OOP, Types of Inheritance in Java, Inheriting Data members and Methods, Role of Constructors in inheritance, Overriding Superclass Methods ,Use of “super” , Polymorphism in inheritance.</p>	8
Module 4	<p>UNIT IV - EXCEPTION HANDLING The Idea behind Exception ,Exceptions & Errors ,Types of Exception,Control Flow In Exceptions, JVM reaction to Exceptions ,Use of try, catch, finally, throw, throws in Exception Handling ,In-built and User Defined Exceptions.</p> <p>UNIT V - ARRAY & STRING Defining an Array, Initializing & Accessing Array, Multi –Dimensional, Array, Operation on String, Mutable & Immutable String.</p> <p>UNIT VI - GUI PROGRAMMING Introduction to Applets and Interfaces, Designing Graphical User Interfaces in Java, Components and Containers.</p>	13

Books Recommended:

1. Java for Programmers, P.J. Dietel, H. M. Dietel, Pearson Education.
2. Java SE 6, Joel Murach, A. Steelman, SPD Pvt. Ltd.
3. Head first java, Kathy Sierra, Bert Bates, Oreilly.
4. Core Java, Cay Horstman and Gary Cornell, Prentice Hall

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (3 rd Semester)	Electronic Devices and Circuits	ECT206	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: None.

Objectives: To understand basics of semiconductors, Diodes, Transistors, Operational amplifiers and their application.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Understand</u> the basics of semiconductor physics, PN junction diode and list of diode applications.	2
CO2	<u>Explain</u> the construction, basics of operation and performance analysis of the devices like BJT, MOSFET.	5
CO3	<u>Develop</u> knowledge of OPAMP and its negative feedback applications	3
CO4	<u>Analyze</u> various feedback amplifiers and positive feedback applications of OPAMP such as oscillators.	4

Details of the Syllabus:

Module No.	Particulars	Hours
Module 1	Introduction to Semiconductors and P-N junction diode: Band theory of Insulators, semiconductors and metals, Intrinsic and extrinsic semiconductors, Current components and IV characteristics of diode, Breakdown, diode circuits: half wave, full wave rectifiers, Filter and regulator circuits, clipping and clamping circuits.	10
Module 2	Transistors: Construction and characteristics of bipolar junction transistors (BJTs)-Common Base, Common emitter, Common Collector configuration, DC load line, Transistor as a switch and Amplifier. Low frequency, h-parameter model, Analysis and design of transistor amplifier circuits using h-parameters., Biasing and stability techniques, MOSFET construction and characteristics.	12
Module 3	Operational Amplifiers: Introduction to Op-amp, Inverting and non-inverting configuration, Applications–adder, subtractor, integrator, differentiator.	10
Module 4	Amplifiers And Oscillators: Classification of amplifiers, concept of feedback, types of feedback, general characteristics of feedback amplifiers, Single stage RC coupled amplifier. Oscillators – Criterion for Oscillation, type of oscillators, Power amplifiers.	10

Recommended Books:

S. No.	Title	Author
1	Semiconductor Physics and Devices	Donald A. Neamen
2	Microelectronics	Sedra & Smith
3	Millman's Electronic Devices and Circuits	Millman Halkias Satiyabrata Jit

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (3 rd Semester)	Data Structures Lab	ITL201	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: C Programming

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	Analyze and apply linked lists for polynomial manipulation.	5
CO2	Explore and implement the various sorting techniques and priority queues using heaps	4
CO3	Apply programming for solving different search trees.	3
CO4	Evaluate various graph traversal techniques and implement their algorithms.	5

Lab Details:

1. Arrays

- Write a program to merge two unsorted arrays.
- Write a program to interchange the largest and smallest number in an array.
- Write a program to fill a square matrix with zeros on the diagonals, 1 on the lower right triangle, and -1 on the lower left triangle.
- In a small company, there are five salesmen. Each salesman is supposed to sell three products. Write a program using 2D array to print (i) the total sales by each salesman and (ii) total sales of each item.
- Write a program to implement a sparse matrix using an array.

2. Linked List

- Write a program that uses functions to perform the following operations on a singly linked list.: i) Creation; ii) Insertion; iii) Deletion; iv) Traversal
- Write a program that uses functions to perform the following operations on a doubly linked list: i) Creation; ii) Insertion; iii) Deletion; iv) Traversal
- Write a program that uses functions to perform the following operations on circular linked list: i) Creation; ii) Insertion; iii) Deletion; iv) Traversal
- Write a program to implement a circular queue using arrays and use it to simulate a producer-consumer problem.
- Write a program to implement a header linked list.
- Write a program to store a polynomial using a linked list. Also, perform addition and subtraction on two polynomials.

3. Stacks

- Write a program to implement stack (its operations) using an array.
- Write a program to implement the stack (its operations) using linked lists (pointers).
- Write a program to check the nesting of parentheses using a stack.

- d. Write a program to convert an infix expression to a postfix expression.
- e. Write a program to convert an infix expression to a prefix expression.

4. Queue

- a. Write a program to implement linear queue (its operations) using array.
- b. Write a program to implement linear queue (its operations) using linked list (pointers).
- c. Write a program to implement priority queue.
- d. Write a program that finds the solution to the Josephus problem using Circular linked list.

5. Trees

- a. Write a program to implement the tree traversal methods (Recursive and Non-Recursive).
- b. Write a program to implement i) Binary binary search tree using linked list.
- c. Write a program to implement single right in-threaded binary tree.
- d. Write a program to implement insertion, deletion, and display (in order, preorder, and postorder) on binary search tree with the information in the tree about the details of an automobile (type, company, year of manufacture).
- e. Write a program to perform the following operations: i) Insert an element into an AVL tree. ii) Delete an element from an AVL tree. iii) Search for a key element in an AVL tree. iv) Count the number of nodes, internal nodes, & external nodes.

6. Multi-way Search Tree

- a. Write a program to implement B Tree using linked list.
- b. Write a program to implement a B+ tree using a linked list.

7. Adjacency List

- a. Write a program to create a graph of n vertices using an adjacency list. Also, use the code to read and print its information, and finally delete the graph.

8. Graph Traversal

- a. Write a program to implement breadth-first search algorithm.
- b. Write a program to implement depth-first search algorithm.

9. Shortest Path Algorithm

- a. Write a program that finds the cost of a minimum spanning tree.
- b. Write a program to implement the modified Floyd warshall algorithm to find the matrix path.
- c. Write a program to implement the Dijkstra algorithm using priority queues.

10. Searching

- a. Write a program that uses both recursive and non-recursive functions to perform the following searching operations for a key value in a given list of integers: i) linear search; ii) binary search.

11. Sorting

- a. Write a program that implements the following sorting methods to sort a given list of integers in ascending order i) Bubble sort; ii) Insertion sort; iii) Selection sort
- b. Write a program that implements the following sorting methods to sort a given list of integers in ascending order i) Merge sort ii) Quick sort iii) Radix sort

12. Topological Sorting

- a. Write a program to implement topological sorting.

13. Heaps

- a. Write a program to implement heap sorting.
- b. Write a program to implement priority queue using heaps.
- c. Write a program to implement a binary search tree using heaps.

14. Hashing and Collision

- a. Write a program to show searching using closed hashing.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (3 rd Semester)	Object Oriented Programming Lab	ITL203	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: C Programming

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Apply</u> programming for problem-solving and use various basic components of object oriented programming.	3
CO2	<u>Implementing</u> the concepts of decision control and loop control structures in Java	3
CO3	<u>Analyze</u> and <u>evaluate</u> the concepts of ArrayList, LinkedList, Queue, multithreading, scheduling, synchronization and classes	5
CO4	<u>Design</u> a graphical user interface in Java	6

Lab Details:

1. Introduction

- a. Download and install the Java Framework

2. Classes and Objects

- a. Programs to understand how to create and use different classes in Java.
- b. Programs to understand differences between logical and arithmetic operators.
- c. Programs to understand differences between logical and bitwise operators.
- d. Programs to evaluate algebraic expressions in C.

3. Constructors and Destructors

- a. Programs to obtain a full understanding of signed, unsigned, long and short numbers in C.
- b. Programs to access members of a class.
- c. programs to create objects inside a class.

4. Methods Overloading

- a. Programs to implement method overloading by changing the number of parameters.

5. Inheritance

- a. Programs to implement inheritance in java.
- b. programs to implement inheritance using method overloading and overriding.

6. Multithreading

- a. Programs to implement multithreading in java.
- b. programs to implement methods of multithreading in java.

7. Package

- a. Programs to implement builtin packages in java.
- b. Programs to implement user defined packages in java.

8. Creating java applets

- a. Develop an applet in Java that displays a simple message.
- b. Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

9. Linked lists

- a. Create a doubly linked list of elements.
- b. Delete a given element from the above list.
- c. Display the contents of the list after deletion.

10. Arrays

- a. Programs to implement declaration and initialization of arrays in java..
- b. Programs to implement looping through arrays in java.
- c. Programs to implement multidimensional arrays.

11. Strings

- a. Programs to create strings in java.
- b. Programs to implement java string operations.

12. Classes

- a. Programs to create classes in java.
- b. Programs to understand java objects.
- c. Programs to access different members of classes.

13. Exception handling

- a. Programs to understand exception handling using try and catch.
- b. Programs to understand exception handling using java finally block.
- c. Programs to understand exception handling using throw and throws keywords.

14. File Handling

- a. Programs to understand creating, reading, writing and deleting a file in java.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (3 rd Semester)	Electronic devices and circuits Lab	ECL207	0-0-2-1
Evaluation Policy	Internal Assessment	External Assessment	Total
	60 Marks	40 Marks	100 Marks

Pre-requisites: Electronic devices and circuits.

Objectives: To acquire knowledge and become familiar with the different characterization techniques to analyse, synthesize basic electronic networks to get desired output.

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

CO No.	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Select</u> and <u>explain</u> the working of different electronic equipment used in Electronics laboratories.	5
CO2	Experimental measures on different types of electronic circuit and <u>analyze</u> their operation under different operating conditions.	5
CO3	<u>Classify</u> relevant information to supplement the Electronic devices and circuits course.	4
CO4	<u>Examine</u> the characterization of diodes, BJT, and OP-AMP for different applications.	4

Details of the Syllabus:

Exp. No.	Particulars
1	Study of CRO-Measurement of voltage frequency and phase of a given waveform.
2	To obtain V-I characteristics of PN Junction diodes.
3	To assemble a half wave and a full wave rectifier and to study their performance.
4	To study Clipping and Clamping circuits with some examples.
5	Comparison of Zener diode and Avalanche diode characteristics and to use Zener diode as a voltage regulator.
6	To obtain transistor characteristics in the following configurations. a) Common base b) Common emitter.
7	To assemble a CE amplifier and observe its performance.
8	Study characteristics of OP-AMP as inverting and non-inverting amplifier.
9	Study applications of OP-AMP as integrator and differentiator.
10	To measure the following parameters of a typical OP-AMP. a. I/P Impedance. b. O/P Impedance. c. Slew rate. d. CMRR.
11	Feedback a. To assemble current series feedback amplifier and study its performance. b. To assemble a voltage shunt feedback amplifier and study its performance.
12	To assemble an RC phase shift oscillator.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (3 rd Semester)	Web Programming	ITL202	1-0-2-2
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: None.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Construct</u> basic websites using HTML and Cascading Style Sheets.	6
CO2	<u>Build</u> dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms.	6
CO3	<u>Design</u> the web applications using React.js	6
CO4	<u>Design, develop, and deploy</u> database-driven web applications	6

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	<p>Introduction to web programming</p> <p>HTML : Introduction to HTML, HTML Basic Formatting Tags, HTML-Grouping Using Div Span, HTML Lists, tables, images, forms, fonts, commenting code, color, hyperlink, iFrame, Headers, XHTML, Meta tags, Character entities, frames and frame sets, Browser architecture and Web site structure.</p> <p>CSS : Need for CSS, introduction to CSS, basic syntax and structure, using CSS, background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, List Tables, Box Model, Display Positioning, Floats</p>	10
Module 2	<p>JavaScript : General overview of JavaScript, Syntax, Variables, Values, Data Types, Data Types, Expressions and Operators, Control structures, Error handling, Numbers, Strings, Arrays, Functions, JSON, Objects, Functions, Classes</p> <p>JS HTML DOM, Browser BOM, Web API's, AJAX, JSON, Graphics</p>	10

Module 3	Introduction to React.js Role of NodeJs and NPM in React, Installation of React plugins in IDE, Installation of Node, and NPM, Installation of React, Data-flow in React, How to create NPM package for React, React Component : use and benefits, How to render view in React, Different Methods to Render view and Components, Es6 with React, JSX, Components with and without JSX, React Props and State, React Prop in practice, React State in practice, React Developer Tool, Event Handling	10
Module 4	React State: Counter App, Handling Lists, Handling User Input, Styling Your Component, How to implement theme in react, create layout based structure in React, Lifecycle methods, Hooks, State Hook, Effect Hook, Building Your Own Hooks, Hooks API Reference, Fetching data Introduction to Django, Introduction to Back-End Web Development using Django, Django Models, Views and Templates, HTML forms in Django templates, form widgets and field customization, Django admin interface, Adding actions and filters in Django admin, Securing Django admin interface, Authentication and Authorization, REST Framework, Testing in Django, Deployment and Project Showcase	12

Books Recommended:

1. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins
2. "JavaScript: The Definitive Guide" by David Flanagan
3. "Learning React: A Hands-On Guide to Building Web Applications Using React and Redux" by Kirupa Chinnathambi
4. "Test-Driven Development with Python, Django, and JavaScript" by Harry J.W. Percival
5. "Django for Professionals" by William S. Vincent

Semester IV

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (4 th Semester)	Operating System	ITT251	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: Computer Programming.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Determining</u> the importance of operating system in the working of the hardware	5
CO2	<u>Apply</u> the process virtualization and coordination mechanisms of the operating system for the effective use of the processor	3
CO3	<u>Evaluate</u> and <u>analyze</u> the memory virtualization and allocation policies of the modern operating systems	5
CO4	<u>Implement</u> and <u>evaluate</u> the file management system implemented by the operating system	5

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	<p>INTRODUCTION TO OPERATING SYSTEMS: Computer System Overview - Basic Elements, Instruction Execution, Instruction set Architecture, Operating system functions and structure - Virtualization, Concurrency, and File system, Evolution of Operating System.</p> <p>PROCESS VIRTUALIZATION: Introduction to System Calls, Process Concept, Implementation of process, Interprocess communication, Process virtualization, System call and Context Switch implementation.</p> <p>PROCESS SCHEDULING: Process Scheduling, Scheduling metrics, Scheduling policies, Multi-Feedback Queue Scheduling, Multiprocessor scheduling.</p>	12
Module 2	<p>MEMORY VIRTUALIZATION: Virtual Memory, Virtual memory implementation using Segmentation, Segmentation implementation, Fragmentation, Virtual Memory Implementation using Paging, Paging Implementation, Page Tables, Page Replacement Policies. Thrashing.</p>	12
Module 3	<p>CONCURRENCY: Critical Section problem, Locks and their implementation, Atomic instructions, Condition Variables, Semaphores, Monitors,</p>	10

	Readers/Writers problem, Concurrency bugs - Deadlocks, Prevention and Avoidance (Safe State and Bankers Algorithm).	
Module 4	FILE SYSTEM I/O management and disk scheduling – I/O devices, organization of I/O functions; OS design issues, I/O buffering, disk scheduling. File management – File System interface and its implementation, FAT and NTFS file system, File system reliability.	8

Books Recommended:

1. A. Silberschatz, P. B. Galvin and G. Gagne “Operating System Principles”.
2. Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau “Operating Systems (Three easy pieces)”.

Other Books Recommended:

1. Andrew S. Tannenbaum & Albert S. Woodhull, “Operating System Design and Implementation”, Prentice Hall.
2. xv6: a simple, Unix-like teaching operating system by Russ Cox Frans Kaashoek Robert Morris
3. Andrew S. Tannenbaum, “Modern Operating Systems”, Prentice Hall.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (4 th Semester)	Software Engineering	ITT252	3-0-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: None.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Choose</u> appropriate process models and design phases for modern software development	6
CO2	<u>Develop</u> proficiency in requirement engineering, software design principles, and object modeling techniques using UML, ensuring effective translation of requirements into comprehensive specifications	6
CO3	<u>Acquire</u> practical skills in debugging, testing, and maintenance processes, employing a variety of techniques such as Black Box Testing, Regression Testing, and Reverse Engineering to ensure software quality and reliability.	3
CO4	<u>Explore</u> software release processes, version management strategies, and project management principles, enabling effective planning, monitoring, and control of software development projects while integrating AI technologies where applicable.	4

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	<p>Introduction to Software Engineering: What is Engineering? ;Glimpses of core engineering disciplines; Why is Software Development is not Construction?; What is Software Engineering?; : Evolution and impact of software engineering</p> <p>Software Development Life Cycle (SDLC) Phases and Models:</p> <p>Phases: Requirements and Analysis; Design, Development; Testing; Deployment; Maintenance</p> <p>Models: Waterfall; Iterative-Incremental; V Shaped; RAD; Prototyping, Evolution Spiral; Agile; SCRUM</p>	8
Module 2	<p>Requirement Engineering: Feasibility study, Functional and non-functional requirements, Requirement gathering, Requirement analysis and specification.</p> <p>Software design: Issues in software design, Modularity; Coupling & Cohesion; Data Flow Diagrams (DFD). Function oriented, software metrics and SCM</p> <p>Object modelling : User interface design, unified process, Object modelling using UML, Use Case, Sequence, Communication (Collaboration), Activity, and State Chart; Structural UML Diagrams: Class and Object. Coding standards and code review techniques.</p>	13

Module 3	<p>Software Debugging Process: Progressive grey-boxing at multiple abstractions; Debugging Life Cycle; Debugging Guidelines.</p> <p>Software Issue / Bug Tracking Process: Sync Issue tracking with SDLC phases and Debugging; Issue / Bug Life Cycle</p> <p>Software Testing Process: Verification and Validation: Black Box Testing, White Box Testing, Grey Box Testing, code coverage, cyclomatic complexity Directed Automated Random Testing (DART),; Regression, Unit, Integration, and System Testing: Alpha, Beta, Acceptance, Testing tools, manual and automatic testing.</p> <p>Software maintenance: maintenance models, Regression testing, Reverse Engineering, Re-Engineering, evolution, Quality Management, Process Improvement, Risk Management.</p>	13
Module 4	<p>Software Release Process: Release Types: Major, Minor, Patch, Hot-fix; Release Numbering: Semantic Versioning; Release Guidelines</p> <p>Software Version Management Process: Branching and Merging; Check-in Guidelines</p> <p>Software Project Management: Project Management: Planning, Monitoring and Control (PMC), Project Estimates: Software Size, Time, Effort Cost, Tools: Gantt Chart, PERT, Resource Histogram; People Management; Program Management</p> <p>Smart Software Engineering: AI in Software Engineering; Software Engineering in AI; AI Engineering</p>	8

Books Recommended:

1. Rajib Mall, Fundamentals of Software Engineering, Prentice Hall India, 2014
2. Pankaj Jalote, An Integrated Approach To Software Engineering, Narosa, 2005
3. Roger S Pressman, Software Engineering: A Practitioner’s Approach, 7th Edition, McGraw Hill Education, 2009
4. Manoj Kumar Lal, Knowledge Driven Development: Bridging Waterfall and Agile Methodologies, Cambridge & IISc Press, 2018

Reference Books:

1. Richard Fairley, "Software Engineering Concepts", McGraw Hill.
2. Stephan Schach, “Software Engineering”, Tata McGraw Hill.
3. Pfleeger and Lawrence, “Software Engineering: Theory and Practice”, Pearson Education.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (4 th Semester)	Digital Electronics and Logic Design	ECT251	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: Electronic devices and circuits.

Objectives: The course will focus on introducing the concepts of logic design and how those concepts can be applied to come up with methodologies that allow the design of digital circuits. The course takes a concrete bottom-up approach starting from the basic number theory and Boolean laws and then building incrementally the design process through logic gates, flip-flops, registers etc. This should allow the student to develop a solid understanding of the basic design process in the digital paradigm. By the end of this course a student will have a more relevant view of the modern digital design field, and a better understanding of how the modern computers and other digital devices function.

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

CO No.	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Understand</u> the concepts of logic design and apply those concepts to formulate design problems as Boolean functions.	2
CO2	<u>Analyze</u> the functionality of the formulated design problems at the gate level to avoid any hazards, thereby ensuring a complete timing closure of the intended circuit.	4
CO3	<u>Choose</u> a proper circuit realization for a given problem and <u>analyze</u> and <u>evaluate</u> the resultant performance in terms of some basic parameters.	5
CO4	<u>Create</u> alternate circuit realizations for a given problem by exploiting the inherent redundancies and analyzing different target elements and platforms.	6

Details of the Syllabus:

Module No.	Particulars	Hours
Module 1	Number System & Boolean Algebra: Binary Number system, Conversion of bases, Complements, Weighted and Non-weighted codes, Binary Arithmetic, Boolean Laws and Logic Gates, Boolean functions- Canonical and Standard forms, Simplification of Boolean Functions, Gate Implementations, Hazards and Glitches, Hazard detection and Hazard free implementations.	12

Module 2	Combinational Logic Design: Design procedure, Adders and Subtractors, Encoders and Decoders, Multiplexers/De-multiplexers and their use in combinational logic design, Digital Comparators, Timing response in Combinational networks.	10
Module 3	Sequential Logic Design: Latches and Flip-flops, Finite State Machines and Controllers, FSM based design- Counters, Sequence detectors, Signal generators, Moore and Mealy machines, Timing in state machines, Registers.	10
Module 4	Digital Logic families: RTL, DCTL, DTL, TTL, ECL, CMOS logic families.	6
Module 5	Programmable Logic devices: PLAs, PALs, CPLDs, FPGAs.	4

Recommended Books:

S. No.	Title	Author
1	Digital Principles and Applications	Donald P. Leach, Albert Paul Malvino and Goutam Saha
2	Digital Logic and Computer Design	M. Morris Mano
3	Modern Digital Electronics	R.P. Jain
4	Digital Design	Frank Vahid
5	Contemporary Logic Design	Randy H. Katz and Gaetano Borriello

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (4 th Semester)	Database Management Systems	ITT253	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: None.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Apply</u> relational database theory and be able to describe relational algebra expression, tuple and domain relational expressions for queries.	3
CO2	<u>Design</u> and <u>construct</u> queries using SQL and analyze the use of normalization and functional dependency.	6
CO3	<u>Apply</u> and <u>analyze</u> the concept of transaction, concurrency control and recovery in the database.	4
CO4	<u>Analyze</u> and <u>evaluate</u> the purpose of query processing and optimization, indexing and hashing techniques used in database design.	5

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	<p>Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS.</p> <p>Database System – Concepts and architecture, Database users and administrator, database types.</p> <p>Introduction to Data Modelling: Data models definition and types, Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model.</p>	10
Module 2	<p>Introduction to relational databases: The relational model -Keys, Relational algebra – Domain relational calculus – Tuple relational calculus – Fundamental operations – Additional operations – SQL fundamentals-QUERIES,</p>	12

	<p>CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active database</p> <p>Introduction to distributed databases and client/server databases.</p> <p>Schema Refinement: Relational database design, Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, First, Second, Third normal forms, BCNF, lossless join decomposition, multivalued dependencies, Fourth normal form, Fifth normal form.</p>	
Module 3	<p>Transaction: Concept, Transaction State, Transaction Recovery, ACID Properties, System Recovery, Media recovery, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability.</p> <p>Locking Protocols: Two phase locking, Intent locking, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity.</p>	10
Module 4	<p>Implementation Techniques: Overview of physical storage media – Magnetic disks, Tertiary storage, File organization –Organization of records in files.</p> <p>Indexing and hashing: Introduction, ordered indices, B trees index files, Static hashing, dynamic hashing, RAID organization and levels.</p> <p>Data warehouse and data mining- basic concepts and overview.</p>	10

Books Recommended:

1. Fundamentals of Database Systems by R. and Navathe, S.B., Pearson Education.
2. Database System Concepts by Silberschatz, Korth, McGraw hill

Other Books Recommended:

1. Database System Concepts by Abraham, H. and Sudershan, S., McGraw-Hill
2. Database Management Systems by Ramakrishnan, R. and Gekhre, J., McGraw-Hill.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
B. Tech. Information Technology Engg. 2 nd Year (4 th Semester)	Introduction to Probability Theory and Statistics	MAT-218	3-0-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: A student should have basic knowledge of set theory.

Course Outcomes: At the end of the course, a students should be able to:

CO NO.	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Analyze</u> various concepts of statistics and apply to various engineering problems	4
CO2	<u>Evaluate</u> various engineering problems using concepts of probability	5
CO3	<u>Solve</u> various engineering problems related to discrete and continuous distributions	3
CO4	<u>Analyze</u> sampling theory and apply it to various engineering problems	4

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	Introduction to basic Statistics, moments, correlation, regression, methods of least square, curve fitting (polynomials, exponentials).	14
Module 2	Basic definitions of probability, conditional probability with standard results, Bays theorem with examples. Discrete and Continuous Random variables, Distribution functions, Expectation and Variance of Probability distribution, and Moment Generating function, Moments and properties.	14
Module 3	Discrete distributions: Binomial, Poisson and Geometric distributions and their applications. Continuous distribution: Uniform, Exponential and Normal distributions, Normal approximation to Binomial distribution and their applications.	6
Module 4	Introduction to sampling theory, types of sampling, purposive sampling, random sampling, simple sampling, stratified sampling, test of significance, null and alternate hypothesis, errors in sampling.	8

Recommended Books:

1. Johnson, Miller and Freund, *Probability and Statistics for Engineers*, Pearson Education India, 8th Edition, (2015).
2. S. C. Gupta, *Fundamentals of Statistics*, 8th Revised Edition, Himalaya Publication, (2023).
3. S. Ross, *A First Course in Probability*, 6th Edition, Pearson Education India, (2002).

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (4 th Semester)	Signals and Systems	ECT257	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: None.

Objectives: To become acquainted with different signal types and their applications in various systems. Characterize Linear Time-Invariant systems, utilize Laplace Transform techniques for signal and system analysis, and apply Discrete Fourier Analysis for signal analysis and LTI system characterization.

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

CO No.	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Understand</u> and <u>classify</u> various types of signals and perform basic operations on them.	4
CO2	<u>Analyze</u> the properties of systems and characterize Linear time invariant systems.	4
CO3	<u>Apply</u> Laplace Transform techniques for analysis of signals and systems.	3
CO4	<u>Understand</u> and <u>apply</u> Discrete Fourier Analysis to analyze signals and characterize LTI systems	3

Details of the Syllabus:

Module No.	Particulars	Hours
Module 1	Introduction to signals: Classification of signals; Deterministic and non-deterministic, periodic and aperiodic, even and odd signals, energy and power signals, elementary signals; exponential, sinusoidal, impulse, step, ramp, pulse, square wave signals. Time shifting, time scaling and time-inversions of signals.	10
Module 2	Linear Time invariant systems: Continuous time system, basic system properties like causality, time invariance, stability, linearity, memory, order of system, interconnection of systems, Linear time invariant systems, characterization, unit impulse response, convolution, properties of LTI systems, linear constant co-efficient differential equations and system description.	12
Module 3	The Laplace Transform: Definition, relation between Laplace and Fourier transforms, region of convergence, properties of Laplace transform, initial and final value theorems, convolution, transfer function of LTI system, concept of poles and zeroes, stability criteria.	10
Module 4	Discrete Fourier analysis of signals and systems: Discrete Fourier transform of aperiodic signals and its properties, Discrete Fourier transform of periodic signals, Discrete time LTI systems characterized by Linear Constant coefficient Difference Equation.	10

Recommended Books:

S. No.	Title	Author
1	Signals & Systems	Simon & Haykins,
2	Signals & Systems - Continuous and Discrete	R.E.Zeimer, W.H.Tranter and R.D.Fannin
3	Principles of Linear Systems and Signals	B. P. Lathi

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (4 th Semester)	Digital Electronics & Logic Design Lab	ECL255	0-0-2-1
Evaluation Policy	Internal Assessment	Final Assessment	Total
	60 Marks	40 Marks	100 Marks

Prerequisite: Digital Electronics & Logic Design.

Objectives: To acquire knowledge and become familiar with the different characterization techniques to analyze, and synthesize the digital logic, combinational and sequential circuits.

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

CO No.	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Identify</u> relevant information to supplement the Digital Electronics & logic Design course.	2
CO2	<u>Develop</u> competence in Combinational Logic Problem identification and solution.	3
CO3	<u>Develop</u> design capability in the field of combinatorial logic using gates and blocks.	3
CO4	<u>Analysis</u> and <u>design</u> of synchronous and asynchronous sequential circuits.	6

Details of the Syllabus:

Exp. No.	Particulars
1	To verify the truth table of following logic gates: AND, OR, NOT, NAND, NOR, XOR and XNOR.
2	To implement XOR and XNOR using universal logic gates.
3	a. To verify DeMorgans law using logic gates. b. To implement typical Boolean expressions and check their equality.
4	To design and realize: - a. Half adder and verify its truth table. b. Full adder and verify its truth table.
5	To design and realize: - a. Half subtractor and verify its truth table. b. Full subtractor and verify its truth table.
6	To design a multiplexer/demultiplexer using two input NAND gates.
7	To design a 4-bit binary to decimal converter.
8	To design a 4-bit binary to gray code converter.
9	To design a modulo-10 counter.
10	Given a frequency f obtain the waveforms with frequencies $f/2$, $f/5$ & $f/10$.
11	Design and realize the following flip-flops using logic gates. a. RS flip-flop. b. JK flip-flop. c. D flip-flop. d. T flip-flop.
12	Use PLL as: a. Frequency multiplier. b. Frequency demodulator.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (4 th Semester)	Operating System Lab	ITL251	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: Computer Programming.

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

COs	Course Outcomes	Bloom's Taxonomy Level
C01	<u>Apply</u> programming for problem-solving and use various basic components of C program.	3
C02	<u>Illustrate</u> the use of functions and various decision and loop control structures.	4
C03	<u>Explore</u> diverse use cases of arrays and strings.	4
C04	<u>Evaluate</u> various data structures like structures and unions and implement dynamic memory allocation and file handling.	5

Lab Details:

1. Introduction

- a. Understanding the working of open source operating system
- b. Working under linux operating system

2. Open Source Operating system

- a. Compiling and installing the xv6 operating system.
- b. Exploring the code of the open source operating system

3. System Calls

- a. Understand the working of the system calls
- b. Implement the user defined system calls.

4. Signal handlers

- a. Understand the working of signal handlers.
- b. Registering the new signal handlers in the system

5. Scheduling Policies

- a. Explore the scheduling policies in the xv6 or in any open source OS
- b. Implement the scheduling policies and test them on a real OS

6. MultiThreading

- a. Implement multithreading programs

7. Synchronization 1

- a. Understand the synchronization issues with the help of multithreading programs
- b. Implement the peterson's solution

8. Synchronization 2

- a. Implement the critical section based multi-thread program.
- b. Implement the locks

9. Synchronization 3

- a. Implement the semaphores
- b. Implement Condition Variables

10. Synchronization 4

- a. Programs to understand the resource allocation to handle deadlocks in the system

11. Memory Management

- a. Programs to understand the memory management
- b. Programs to dynamically allocate memory and analyze the same

12. Page Replacement Policies

- a. Programs to understand the working of page replacement policies
- b. Programs to analyze the effect of page replacement policies on the performance

13. File system

- a. Understand the file system of some open source OS
- b. Implement a basic file system

14. Disk Scheduling

- a. Programs to implement the disk scheduling algorithms

Year (Semester)	Course Title	Course Code	L-T-P-Credits
2 nd Year (4 th Semester)	Database Management Systems Lab	ITL253	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: None.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Analyze</u> the database using queries to retrieve records.	4
CO2	<u>Evaluate</u> PL/SQL for processing databases.	5
CO3	<u>Formulate</u> query, using SQL, solutions to a broad range of queries, and data update problems.	6
CO4	<u>Develop</u> solutions using database concepts for real-time requirements.	3

Lab Details:

1. Creation of Tables

- a. Create a table called Employee with the following Structure and Answer the following queries.

Name	Type
Empno	Number
Ename	Varchar2 (20)
Job	Varchar2 (20)
Mgr	Number
Sal	Number

i) Add a column commission with domain to the Employee table.

ii) Insert any five records in to the table.

```
Sql> INSERT INTO Employee VALUES (1, 'King', 'ITmanager', '100', '20000');
```

```
Sql> INSERT INTO Employee VALUES (5, 'blake', 'IT', '200', '30000');
```

```
Sql> INSERT INTO Employee VALUES (9, 'raj', 'manager', '300', '40000');
```

```
Sql> INSERT INTO Employee VALUES (19, 'clarke', 'Assistant', '400', '50000');
```

- Sql> INSERT INTO Employee VALUES (25, 'mohan', 'clerk', '500', '60000');
- iii) Update the column details of job.
 - iv) Rename the column of Employ table using alter command.
 - v) Delete the employee whose empno is 19.

2. Queries using DDL

- i) CREATE ii) DROP iii) ALTER iv) RENAME v) TRUNCATE

- a) Create a table EMPLOYEE with following schema: (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name, Job_id, Salary)
- b) Add a new column; HIREDATE to the existing relation.
- c) Change the datatype of JOB_ID from char to varchar2.
- d) Change the name of column/field Emp_no to E_no.
- e) Modify the column width of the job field of emp table.

3. Queries using DML

- i) INSERT ii) UPDATE iii) DELETE

- a) Create a table EMPLOYEE with following schema: (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name, Job_id, Salary)
- b) Insert atleast 5 rows in the table.
- c) Display all the information of EMP table.
- d) Display the record of each employee who works in department D10.
- e) Update the city of Emp_no-12 with current city as Nagpur.
- f) Display the details of Employee who works in department MECH.
- g) Delete the email_id of employee James.
- h) Display the complete record of employees working in SALES Department.

4. Queries using aggregate functions

- i) Number Function ii) Aggregate Function iii) Character Function iv) Conversion Function v) Date Function

- a) Create a table EMPLOYEE with following schema: (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name, Job_id, Designation, Salary)
- b) List the E_no, E_name, Salary of all employees working for MANAGER.
- c) Display all the details of the employee whose salary is more than the Sal of any IT PROFF.
- d) List the employees in the ascending order of Designations of those joined after 1981.
- e) List the employees along with their Experience and Daily Salary.
- f) List the employees who are either 'CLERK' or 'ANALYST' .
- g) List the employees who joined on 1-MAY-81, 3-DEC-81, 17-DEC-81, 19-JAN-80 .
- h) List the employees who are working for the Deptno 10 or 20.
- i) List the Enames those are starting with 'S' .
- j) Display the name as well as the first five characters of name(s) starting with 'H'
- k) List all the emps except 'PRESIDENT' & 'MGR' in asc order of Salaries.

5. Implementation of different types of operators in SQL

- i) Arithmetic Operator ii) Logical Operator iii) Comparison Operator iv) Special Operator
- v) Set Operator
 - a) Display all the dept numbers available with the dept and emp tables avoiding duplicates.
 - b) Display all the dept numbers available with the dept and emp tables.
 - c) Display all the dept numbers available in emp and not in dept tables and vice versa.

6. Implementation of different types of Joins

- i) Inner Join ii) Outer Join iii) Natural Join..etc

Consider the following schema: Sailors (sid, sname, rating, age) Boats (bid, bname, color)
Reserves (sid, bid, day(date))

- a) Find all information of sailors who have reserved boat number 101.
- b) Find the name of boat reserved by Bob.
- c) Find the names of sailors who have reserved a red boat, and list in the order of age.
- d) Find the names of sailors who have reserved at least one boat.
- e) Find the ids and names of sailors who have reserved two different boats on the same day.
- f) Find the ids of sailors who have reserved a red boat or a green boat.
- g) Find the name and the age of the youngest sailor.
- h) Count the number of different sailor names.
- i) Find the average age of sailors for each rating level. 10. Find the average age of sailors for each rating level that has at least two sailors.

7. Study & Implementation of

- i) Group by & Having Clause ii) Order by Clause iii) Indexing

Create a relation and implement the following queries.

- a) Display total salary spent for each job category.
- b) Display lowest paid employee details under each manager.
- c) Display number of employees working in each department and their department name.
- d) Display the details of employees sorting the salary in increasing order.
- e) Show the record of employee earning salary greater than 16000 in each department.
- f) Write queries to implement and practice the above clause.

8. Study & Implementation of

- i) Sub queries ii) Views

Consider the following schema: Sailors (sid, sname, rating, age) Boats (bid, bname, color)
Reserves (sid, bid, day(date)).

- a) Find all information of sailors who have reserved boat number 101.
- b) Find the name of boat reserved by Bob.
- c) Find the names of sailors who have reserved a red boat, and list in the order of age.
- d) Find the names of sailors who have reserved at least one boat.
- e) Find the ids and names of sailors who have reserved two different boats on the same day.
- f) Find the ids of sailors who have reserved a red boat or a green boat.
- g) Find the name and the age of the youngest sailor.
- h) Count the number of different sailor names.

- i) Find the average age of sailors for each rating level.
- j) Find the average age of sailors for each rating level that has at least two sailors.

9. Study & Implementation of different types of constraints

Create a table called EMP with the following structure.

Name	Type
Empno	Number(6)
Ename	Varchar2 (20)
Job	Varchar2 (10)
Deptno	Number(3)
Sal	Number(7, 2)

- a) Add constraints to check, while entering the empno value (i.e) empno > 100.
- b) Define the field DEPTNO as unique.
- c) Create a primary key constraint for the table(EMPNO).
- d) Write queries to implement and practice constraints.

10. Study and Implementation of

i) Database Backup & Recovery Commands. ii) Rollback, Commit, Save point.

- a) Write a query to implement the save point.
- b) Write a query to implement the rollback.
- c) Write a query to implement the commit.

11. Creating Database/ Table Space

i) Managing Users: - Create User, Delete User ii) Managing Passwords iii) Managing roles: - Grant , Revoke.

- a) Create user and implement the following commands on relation (Emp and Dept).
- b) Develop a query to grant all privileges of employees table into departments table.
- c) Develop a query to grant some privileges of employees table into departments table.
- d) Develop a query to revoke all privileges of employees table from departments table.
- e) Develop a query to revoke some privileges of employees table from departments table.

12. PL/SQL

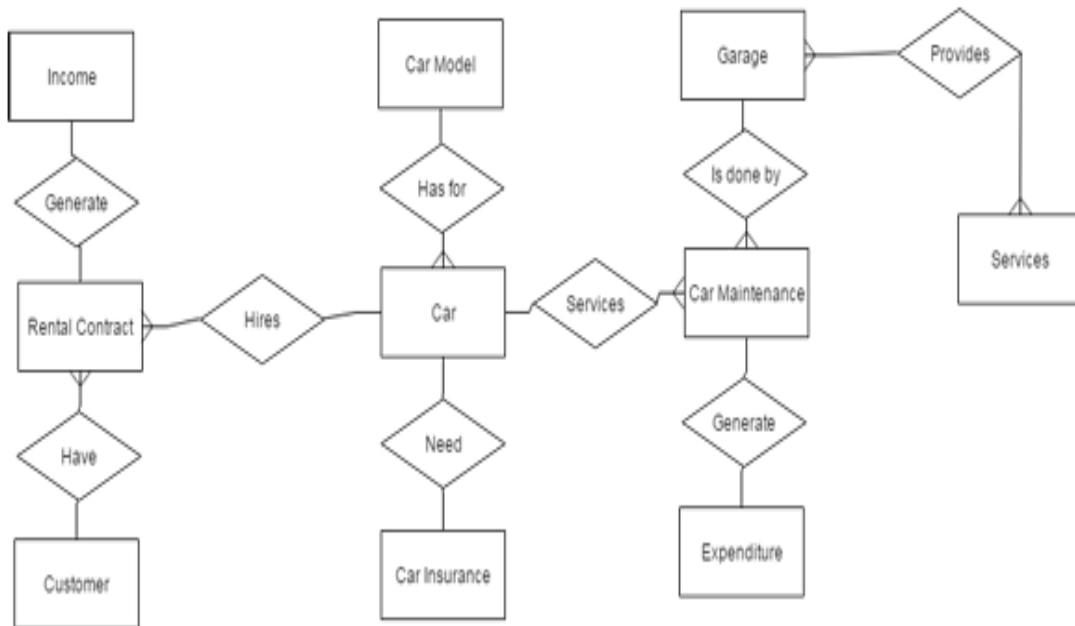
- a) Study of PL/SQL block.
- b) Creation of procedures.
- c) Creation of database triggers, cursors, and functions.

13. CASE STUDY: CAR RENTAL COMPANY

Aim: A database is to be designed for a Car Rental Co. (CRC). The information required includes a description of cars, subcontractors (i.e. garages), company expenditures, company revenues and customers. Cars are to be described by such data as: make, model, year of production, engine size, and fuel type, number of passengers, registration number, purchase price, purchase date, rent price and insurance details. It is the company policy not to keep any car for a period exceeding one year.

All major repairs and maintenance are done by subcontractors (i.e. franchised garages), with whom CRC has long-term agreements. Therefore, the data about garages to be kept in the database includes garage names, addresses, range of services and the like. Some garages require payments immediately after a repair has been made; with others, CRC has made arrangements for credit facilities. Company expenditures are to be registered for all outgoings connected with purchases, repairs, maintenance, insurance etc.

Similarly, the cash inflow coming from all sources - car hire, car sales, insurance claims - must be kept on file. CRC maintains a reasonably stable client base. For this privileged category of customers, special credit card facilities are provided. These customers may also book in advance a particular car. These reservations can be made for any period of time up to one month. Casual customers must pay a deposit for an estimated time of rental, unless they wish to pay by credit card. All major credit cards are accepted. Personal details (such as name, address, telephone number, driving license, number) about each customer are kept in the database.



14. Mini Project

- a) Inventory Control System.
- b) Material Requirement Processing.
- c) Hospital Management System.
- d) Railway Reservation System.
- e) Personal Information System.
- f) Web Based User Identification System.
- g) Timetable Management System.
- h) Hotel Management System.

Semester V

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (5 th semester)	Design & Analysis of Algorithms	ITT301	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: Data Structures

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Evaluate</u> the concept's and complexity of algorithm and <u>analyze</u> different sorting algorithm	5
CO2	<u>Apply</u> the concept of dynamic programming in designing efficient algorithms.	3
CO3	<u>Analyze</u> the Greedy Algorithms.	4
CO4	<u>Evaluate</u> the concept of NP-Complete Problem in real world applications.	5

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	<p>Introduction: Asymptotic notations for time and space complexity, Big-Oh notation, Θ notation, Ω notation, the little-oh notation, the little-omega notation,</p> <p>Recurrence relations: iteration method, recursion tree method, substitution method, master method (with proof), subtract and conquer master method (with proof), Data Structures for Disjoint Sets, Medians and Order statistics.</p> <p>Divide Conquer methods: Binary search, Quick sort, Strassen's algorithm for matrix multiplication, analysis of divide and conquer runtime reference relations.</p>	10
Module 2	<p>Dynamic Programming: Ingredients of dynamic programming, emphasis on optimal substructure, overlapping substructures, memorization. Matrix Chain Multiplication, Longest common subsequence and optimal binary search trees problems, 0-1 knapsack problem, Binomial coefficient computation through dynamic programming. Floyd Warshall algorithm.</p>	12
Module 3	<p>Greedy Algorithms: Elements of Greedy strategy, overview of local and global optima, matroid, Activity selection problem, Fractional Knapsack problem, Huffman Codes, A task scheduling problem.</p> <p>Minimum Spanning Trees: Kruskal's and Prim's Algorithm.</p>	12

	<p>Single source shortest path: Dijkstra and Bellman Ford Algorithm(with proof of correctness of algorithms).</p> <p>String Matching: The naïve String Matching algorithm, The Rabin-Karp Algorithm, String Matching with finite automata, The Knuth-Morris Pratt algorithm.</p>	
Module 4	<p>Tractable and Intractable Problems: NP-Complete Problem, Polynomial-time verification, NP-Completeness and Reducibility, NP-Completeness Proof, NP-hard, Case study of NP-Complete problems (vertex cover problem, clique problem).</p>	8

Books Recommended:

1. Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein, MIT Press.
2. Algorithm design, by Kleinberg, J., & Tardos, E, Pearson Education India.

Other Books Recommended:

1. Algorithms, by Dasgupta, Papadimitrou and Vazirani, McGraw-Hill Education, 2006.
2. Computer Algorithms, by Horowitz, Sahni, and Rajasekaran, Silicon Press, 2007.
3. Algorithm Design, by Kleinberg and Tardos, Pearson, 2005.
4. Algorithm Design, by Goodrich and Tamassia, Wiley, 2001.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (5 th semester)	Artificial Intelligence	ITT 302	3-0-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: None.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Apply</u> basic AI techniques for search techniques.	3
CO2	<u>Apply</u> basic principles in problem solving, inference, perception, knowledge representation and learning.	3
CO3	<u>Evaluate</u> the machine learning concepts for designing a better AI systems	5
CO4	<u>Investigate</u> applications of AI techniques in intelligent agents, expert systems, machine learning models.	4

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	INTRODUCTION: Introduction to AI and intelligent agents. Problem Solving : Solving Problems by Searching, heuristic search techniques, constraint satisfaction problems, stochastic search methods, adversarial search, Game playing : minimax, alpha-beta pruning.	8
Module 2	KNOWLEDGE REPRESENTATION AND REASONING: Building a Knowledge Base : Propositional logic, first order logic, Theorem Proving in First Order Logic. Production Systems, Semantic Nets, Frames and Scripts Formalisms. Resolution in Predicate Logic, Unification, Strategies for Resolution by Refutation. Knowledge Acquisition and learning: Learning from examples and analogy, Rote learning, Neural Learning, Integrated Approach. Planning, partial order planning. Uncertain Knowledge and Reasoning, Probabilities, Bayesian Networks.	10
Module 3	INTRODUCTION TO MACHINE LEARNING: Machine Learning, Supervised Learning, Unsupervised Learning, Reinforcement Learning. Introduction to Probability, Basics Linear Algebra, Statistical Decision Theory – Regression & Classification, Bias – Variance, Overfitting and complexity; training, validation, test data.	16

	Basics of Neural Networks: Multilayer Neural Network, Gradient Descent learning, Back propagation, Empirical Risk Minimization, regularization, Radial Basis Neural Network, Backpropagation through time	
Module 4	<p>EXPERT SYSTEM: Existing Systems (DENDRAL, MYCIN), domain exploration, Meta Knowledge, Expertise Transfer, Self Explaining System. Fuzzy logic: Membership functions, Fuzzy logic, Fuzzy rule generation, Defuzzification, Time dependent fuzzy logic, Temporal fuzzy logics, Case study-to use fuzzy logic for processes control problem</p> <p>Programming Language: Introduction to programming Language- LISP, PROLOG</p>	8

Books Recommended:

1. Rich & Knight, "Artificial Intelligence".
2. Elamie, "Artificial Intelligence", Academic Press.

Other Books Recommended:

1. Char nick "Introduction to Artificial Intelligence", Addison Wesley.
2. Winston, "LISP", Addison Wesley.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (5 th Semester)	Computer Organization and Architecture	ITT303	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: Computer Programming.

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

COs	Course Outcomes	Bloom's Taxonomy Level
C01	<u>Manipulate</u> the representation of numbers inside the computer and <u>evaluate</u> their low level operations	5
C02	<u>Demonstrate</u> how to assemble a simple computer with hardware design including instruction format, instruction set, memory, arithmetic and logical unit, control unit and data.	3
C03	<u>Design</u> a processor pipeline.	6
C04	<u>Evaluate</u> cost performance and <u>design</u> trade-offs in designing and constructing a computer processor including memory	5

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	INTRODUCTION TO COMPUTER ORGANIZATION AND ARCHITECTURE: Computer organization and architecture, Computing Models, Instruction Set Architecture, Tradeoffs in the architecture design, Microarchitecture, Addressing modes. COMPUTER ARITHMETIC: Representation of numbers, 1's complement and 2's complement representation, Design of arithmetic circuits for addition, subtraction, multiplication and division, Floating point number representation, Arithmetic operations for floating point numbers.	10
Module 2	DIGITAL LOGIC DESIGN: PMOS and NMOS transistors, Logic gates using transistors, Combinational and Sequential Logic, Design of registers, SRAM and DRAM memories and their design, PLA INSTRUCTION SET ARCHITECTURE - RISCV (instructor can teach any other relevant ISA): Introduction to the open source ISA, Instruction formats available in the ISA, Instructions of the ISA, Program implementation using the ISA, Compilation flow.	10
Module 3	PROCESSOR DESIGN: Instruction execution, Design of Datapath - Design of Different Stages of the processor including fetch, decode, memory access and register writeback stage, Design of Control path - Design of control unit, Microprogrammed control and Hardwired control	12

	PROCESSOR PIPELINE AND PIPELINE HAZARDS: Single cycle and Multi cycle processors, Processor Pipelining, Performance Equation, Pipeline Hazards, Mechanisms to handle Hazards including forwarding, branch prediction.	
Module 4	<p>MEMORY SYSTEM: Memory organization, Types of memories and performance considerations, organization of memory modules, associative memory, cache memory and related mapping and replacement policies, virtual memory and its implementation.</p> <p>MULTIPROCESSOR SYSTEMS: Introduction to multiprocessors, Symmetric and Asymmetric multiprocessing, FLYNNs classification, Coherence and consistency, Cache Coherence, Basic introduction to Interconnection networks</p>	10

Books Recommended:

1. "Computer Organization and Architecture" by Smruti R. Sarangi.
2. "Digital Design and Computer Architecture" By Harris et al.
3. "Computer Organization and Embedded Systems" by Carl Hamacher, Z. Vranesic, s. Zaky, and N. Manjikian.
4. "Computer System Architecture" by M. Mano

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (5 th Semester)	Theory of Computation	ITT304	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: Computer Programming.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Design</u> finite automata for different computation problems	6
CO2	<u>Design</u> and <u>analyze</u> the nature of grammars for the construction of push down automata	6
CO3	<u>Apply</u> the design of turing machines to recognize the power of languages	3
CO4	<u>Evaluate</u> the complexity of computational problems	5

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	INTRODUCTION: Complexity of computations, automata, computability, complexity, mathematical notations and terminology, definitions, theorems and proofs, types of proofs FINITE AUTOMATA & REGULAR LANGUAGES: Finite Automata, Deterministic and non-deterministic automata, regular operations, regular expressions, regular and non-regular languages, Equivalence of NFA and DFA, Reduction of finite automata.	12
Module 2	GRAMMARS & CONTEXT FREE LANGUAGES: Grammars and types of grammars, Context free grammars, Parsing and ambiguity, Simplification of grammars, Normal forms, Chomsky form, Greibach Normal form, Context Free Languages, Pushdown Automata, closure properties, concepts in parsing, Computability theory	14
Module 3	COMPUTABILITY THEORY: Turing machines, variants of Turing machines, the definition of algorithm Decidability, reducibility, enumerability, Recursive and Recursively Enumerable Languages, Church Turing thesis, Rice's theorem.	10
Module 4	COMPLEXITY THEORY: Time complexity, space complexity, Complexity class P and NP, NP complete problems.	6

Books Recommended:

1. Peter Linz, "An Introduction to Formal Language and Automata", Narosa Publishing house.
2. M.Sipser; Introduction to the Theory of Computation; Singapore: Brooks/Cole, Thomson Learning.
3. John.C.martin, "Introduction to the Languages and the Theory of Computation", Tata McGrawHill.
4. K.Krithivasan and R.Rama; Introduction to Formal Languages, Automata Theory and Computation; Pearson Education.
5. J.E.Hopcroft, R.Motwani and J.D.Ullman , "Introduction to Automata Theory Languages and computation", Pearson Education Asia.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (5 th semester)	Design & Analysis of Algorithms Lab	ITL301	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: Computer Programming, Data Structures.

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

COs	Course Outcomes	Bloom's Taxonomy Level
C01	<u>Apply</u> programming for solving problems based on recurrence relations.	3
C02	<u>Explore</u> diverse use cases of divide and conquer.	4
C03	<u>Implement</u> greedy algorithms for minimum cost spanning tree, Knapsack problem, single source the shortest paths.	3
C04	<u>Illustrate</u> the use of backtracking in solving 8 queens problems.	4

Lab Details:

1. Recurrence Relations

- Design and implement an algorithm that multiplies two 'n' digit numbers faster than $O(n^3)$.
- Design and implement an algorithm that will find the top and the least scores of students from an online quiz. Note: The scores are stored in an array.

2. Sorting

- Write a program to perform Bubble sort for any given list of numbers.
- Write a program to perform Insertion sort for any given list of numbers.
- Write a program to perform Quick Sort for the given list of integer values.

3. Divide and Conquer

- Sort a given set of n integer elements using **Quick Sort** method and compute its time complexity. Run the program for varied values of $n > 5000$ and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the divide and-conquer method works along with its time complexity analysis: worst case, average case and best case.
- Sort a given set of n integer elements using **Merge Sort** method and compute its time complexity. Run the program for varied values of $n > 5000$, and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the divide- and-conquer method works along with its time complexity analysis: worst case, average case and best case.

- c. Write a program to find **Maximum and Minimum** of the given set of integer values.

4. Dynamic Programming

- a. Write a program to implement the All-Pairs Shortest Paths problem using **Floyd's** algorithm.
- b. Write a program to implement the **0/1 Knapsack** problem
- c. Write a program to implement **the Travelling SalesPerson** problem.

5. Greedy Programming

- a. Write a program to implement the **0/1 Knapsack** problem using greedy technique.
- b. Write a java program to implement greedy algorithms for **job sequencing** with deadlines.

6. Minimum Cost Spanning Tree

- a. Find the Minimum Cost Spanning Tree of a given connected undirected graph using **Prim's algorithm**.
- b. Find the Minimum Cost Spanning Tree of a given connected undirected graph using **Kruskal's algorithm**. Use Union-Find algorithms in your program.

7. Single Source Shortest Path

- a. From a given vertex in a weighted connected graph, find the shortest paths to other vertices using **Dijkstra's algorithm**.
- b. From a given vertex in a weighted connected graph, find the shortest paths to other vertices using **Bellman-Ford algorithm**.

8. Priority Queue

- a. Write a program to implement **Huffman Coding** using priority queue.
- b. Write a program to implement **Dijkstra's algorithm** using priority queue.

9. Graph Searching

- a. Write a program to implement the **BFS** algorithm for a graph.
- b. Write a program to implement the **DFS** algorithm for a graph.

10. Binary Search

- a. Write a Program to perform **Binary Search** for a given set of integer values, recursively and non- recursively.

- b. Write a program to implement a Dynamic Programming algorithm for the **Optimal Binary Search Tree Problem**.

11. Backtracking

- a. Given a 8x8 chess board, you must place 8 queens on the board so that no two queens attack each other. Print all possible matrices satisfying the conditions with positions with queens marked with '1' and empty spaces with '0'. Write a program to solve the **8 queens problem** using backtracking.
- b. Write a program to implement a backtracking algorithm for the **Knapsack** problem.

12. Hamiltonian Circuit Problem

- a. Write a program to find all **Hamiltonian Cycles** in a connected undirected Graph G of n vertices using backtracking principle.

13. Branch and Bound

- a. Write a program to generate a **binary string of length N** using Branch and Bound.
- b. Write a program to implement **the Job Assignment Problem** using Branch and Bound.
- c. Write a program to implement the Travelling Salesman Problem using reduced matrix method.

14. Subset Problem

- a. Design and implement a program to find a subset of a given set $S = \{S_1, S_2, \dots, S_n\}$ of n positive integers whose SUM is equal to a given positive integer d. For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$, there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$. Display a suitable message, if the given problem instance doesn't have a solution.
- b. Write a program to solve the **Sum of subsets** problem for a given set of distinct numbers.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (5 th semester)	Artificial Intelligence Lab	ITL302	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: None.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Elicit</u> , <u>analyze</u> , <u>design</u> and <u>implement</u> search strategies	6
CO2	<u>Design</u> and <u>implement</u> game playing and CSP techniques	6
CO3	<u>Design</u> and <u>Develop</u> logical reasoning systems	6
CO4	<u>Design</u> and <u>Develop</u> probabilistic reasoning systems	6

Detailed Syllabus:

Contents	Duration
<p>List of Experiments</p> <ol style="list-style-type: none"> 1. Implement basic search strategies – 8-Puzzle Problem. 2. Implement basic search strategies – 8-Queens Problem 3. Implement basic search strategies – Crypt arithmetic. 4. Implement Best First Search algorithm 5. Implement Genetic algorithm 6. Implement A*, AO* algorithms. 7. Implement Minimax algorithm for game playing 8. Implement Hill Climbing algorithm 9. Solve constraint satisfaction problems 10. Implement Propositional Model checking algorithms. 11. Implement Forward Chaining algorithm. 12. Implement backward chaining algorithm 13. Implement Naïve Bayes Models. 14. Implement Bayesian Networks and perform inferences. 	14 weeks

Books Recommended:

1. Rich & Knight, "Artificial Intelligence".
2. Elamie, "Artificial Intelligence", Academic Press.

Other Books Recommended:

1. Char nick "Introduction to Artificial Intelligence", Addison Wesley.
2. Winston, "LISP", Addison Wesley.

Semester VI

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (6 th Semester)	Computer Networks	ITT351	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: Data Communication

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Discover</u> the need for networking and various error, and access control strategies in the TCP/IP reference model.	3
CO2	<u>Analyze</u> the different flow control strategies and switching techniques.	4
CO3	<u>Compare</u> IP addressing, subnetting techniques, and various routing protocols used at the Network layer.	5
CO4	<u>Review</u> session management mechanism and <u>evaluate</u> various strategies for reducing network congestion and application layer services.	5

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	Introduction: Need for computer networks, layering and protocols. Basics of physical and data link layer: Encoding, Error detection and correction. Access control protocols: Aloha, CSMA, CSMA/CD, CSMA/CA and other controlled-access and channelization-based protocols, Framing, HDLC	8
Module 2	Flow control: Stop and Wait protocols, Sliding Window, Go-Back N and selective repeat. Switching Theory: Circuit Switching, Message switching, Packet switching.	10
Module 3	Network Layer: Internet Protocol, IPv4, IPv6, ARP, DHCP, ICMP, Routing algorithms: Distance vector, Link state, Metrics, Inter-domain routing. Dijkstra, Bellman-Ford Algorithms. Subnetting, Classless addressing, Network Address Translation.	18
Module 4	Transport layer: UDP, TCP. Connection establishment and termination, sliding window revisited, flow and congestion control, timers, retransmission, TCP extensions. Application layer: DNS, SMTP, IMAP, HTTP, etc.	6

Recommended Books:

1. JF Kurose, KW Ross. Computer Networking: A Top-Down Approach
2. Behrouz A. Forouzan. Data communications and Networking
3. William Stallings. Data & Computer Communications, PHI
4. Andrew Tanenbaum. Computer Networks, PHI

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (6 th Semester)	Machine Learning	ITT352	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: Computer Programming, Introduction to Probability and Statistics

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	Apply supervised learning techniques to learn the data dependencies	3
CO2	Evaluate the effect of learning techniques on the overall learning performance	5
CO3	Apply the artificial intelligence techniques to the classification of images	3
CO4	Design the schemes to reduce the amount of data without the loss of information	6

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	<p>INTRODUCTION: Machine Learning, Supervised Learning, Unsupervised Learning, Reinforcement Learning. Introduction to Probability, Basics Linear Algebra, Statistical Decision Theory – Regression & Classification, Bias – Variance, Overfitting and complexity; training, validation, test data.</p> <p>REGRESSION AND CLASSIFICATION: Regression:-Linear Regression-Cost Function, Gradient Descent with single and multivariable, Normal Equation, Regularization Techniques (LASSO), Polynomial Regression Classification:- Logistic Regression-Hypothesis Representation, Decision boundary, Cost Function, Multi class Classification</p>	12
Module 2	<p>PROBABILISTIC MODELLING: Probability Review, Random Variables, Probability Distributions and their relation with Machine Learning models, Gaussian Discriminant Analysis, Naive Bayes</p> <p>NON-PARAMETRIC MODELS: Non-parametric models, K-Nearest Neighbour, Feature Maps, Kernel and Kernel Trick, Support Vector Machine,</p>	12
Module 3	<p>ARTIFICIAL NEURAL NETWORKS: Introduction to neural networks, Simple Perceptron, Activation Functions, Training neural networks using backpropagation, Regularizing neural networks</p>	10
Module 4	<p>UNSUPERVISED LEARNING & REINFORCEMENT LEARNING: Clustering using K-Means, K-Means Issues, Covariance Matrix and its Eigen</p>	8

	vectors, Dimensionality reduction using Principal Component Analysis, Introduction to reinforcement learning, Value Function, Q Learning, DQN.	
--	--	--

Books Recommended:

3. Probabilistic Machine Learning: An Introduction. Kevin Murphy.
4. Pattern Recognition and Machine Learning. Christopher Bishop.
5. Pattern Classification. Richard Duda, Peter Hart and David Stock.
6. Machine Learning. Tom Mitchell.
7. Hands-On Machine Learning with Scikit-Learn and TensorFlow. Aurélien Géron

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (6 th Semester)	Big Data	ITT353	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: Database Management System

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Explore</u> the fundamental concepts of big data analytics.	4
CO2	<u>Analyze</u> various classification algorithms.	4
CO3	<u>Demonstrate</u> use of programming tools PIG & HIVE in the Hadoop ecosystem.	3
CO4	<u>Analyze</u> and <u>Evaluate</u> tools and technologies in Unsupervised Data Analytics.	5

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	<p>Introduction: Big Data Overview, Introduction to the Big Data problem. Current challenges, trends, and applications, Algorithms for Big Data analysis. Data sets, Mining and learning algorithms that deal with large datasets Technologies for Big Data management. Big Data technology and tools, special consideration made to the Map-Reduce paradigm and the Hadoop ecosystem.</p> <p>Data Science: What is data sciences, The rising and importance of data sciences, Big data analytics in industry verticals-Intelligent data analysis–Nature of Data-Analytic Processes and Tools-Analysis vs Reporting Data, Analytics Lifecycle and methodology.</p>	10
Module 2	<p>Modelling: Data Understanding, Data Preparation, Evaluation, Communicating results, Deployment, Data exploration & preprocessing.</p> <p>Measures and Evaluation: Data Analytics: Theory & Methods, Supervised Learning: Artificial Neural Network, classifying with k-Nearest Neighbor classifier, Support vector machine classifier, Decision Tree classifier, Naive Bayes classifier, Bagging, Boosting, Improving classification with the Ada Boost meta algorithm.</p>	10
Module 3	<p>Hadoop: History of Hadoop, Apache Hadoop, Analysing Data with Unix tools, Analysing Data with Hadoop, Hadoop Streaming, Hadoop Echo System</p> <p>HDFS (Hadoop Distributed File System): The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives, Hadoop I/O: Compression, Serialization, Avro and File-Based Data structures.</p>	12

Module 4	<p>Map Reduce: Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features.</p> <p>Unstructured Data Analytics: Technologies & tools, Text mining, Web mining, Operationalizing an Analytics project, Data Visualization Techniques, Creating final deliverable.</p>	10
-----------------	---	-----------

Books Recommended:

1. Big Data: A Revolution That Will Transform How We Live, Work, and Think by Viktor Mayer-Schönberger , Kenneth Cukier.
2. Hadoop: The Definitive Guide by Tom White (Goodreads Author).
3. Real-Time Big Data Analytics: Emerging Architecture [Kindle Edition], Mike Barlow

Other Books Recommended:

1. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.
2. Jay Liebowitz, "Big Data and Business Analytics" Auerbach Publications, CRC press (2013).
3. Tom Plunkett, Mark Hornick, "Using R to Unlock the Value of Big Data: Big Data Analytics with Oracle R Enterprise and Oracle R Connector for Hadoop", McGraw-Hill/Osborne Media (2013), Oracle press.
4. Anand Rajaraman and Jeffrey David Ulman, "Mining of Massive Datasets", Cambridge University Press, 2012.

Engineering Economics and Management (HST004)	Year & Semester: B. Tech 3rd -6th Sem		Total Course Credit: 3		
			L	T	P
	3	0	0		
Evaluation Policy	Mid-Term (26 Marks)	Continuous Assessment (24 Marks)	End-Term (50 Marks)		

COURSE OUTCOMES:

After completing this course, the student must demonstrate the knowledge and ability to:

CO No.	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Apply</u> the management concepts within the engineering domain.	3
CO2	<u>Analyze</u> the various functions of management	4
CO3	<u>Appraise</u> the importance of economics in engineering context	5
CO4	<u>Discuss</u> the operation of different forms of markets and their competitive strategies.	6

Detailed Syllabus:

Module-1	Basics of Management: Definition, Functions, Process and Significance of Management; Evolution of management thought; Theories of Management	10 Hrs
Module-2	Functions of Management: Planning; Organizing; Staffing; Directing and Controlling. Nature and Significance of functions of management	10 Hrs
Module-3	Basics of Economics and Consumer Behaviour: Economics-Meaning, Divisions and Importance (Engineering Context); Demand-Concept and Law, Elasticity of demand-types, measurement, and importance.	10 Hrs
Module-4	Market and Economic Policies: Markets- Features, type; Perfect Competition, Monopoly, Monopolistic Competition, Duopoly and Oligopoly; Introduction to concepts of inflation, GDP, fiscal policy and monetary policy.	12 Hrs

Books Recommended:

Text books	1.	Ahuja, H. L. (2016). Advanced Economic Theory: Microeconomic analysis (Ed. 21st). S. Chand.
	2.	Keat, P. G., College, d., Erfle, S., Banerjee, S., & Young, P. K. Y. (2018). Managerial Economics (Ed. 7th). Pearson.
	3.	Dessler, G., & Varrkey, B. (2020). Human Resource Management (16th ed.). Pearson.
	4.	Koontz, H., Weihrich, H., & Cannice, M. V. (2020). Essentials of Management (Ed. 11th). McGraw Hill, New Delhi.

Referenc e books	5.	Pindyck, R. S., Rubinfeld, D. L., & Banerjee, S. (2022). <i>Microeconomics</i> (Ed. 9 th). Pearson Education.
	6.	Koontz, H., Wehrich, H., & Cannice, M. V. (2020). <i>Essentials of Management</i> (11th ed.). McGraw-Hill.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (6 th Semester)	Computer Networks Lab	ITL351	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: None.

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Implement</u> basic network utilities and analyze network traffic using Wireshark tool.	3
CO2	<u>Configure</u> internetworking devices like routers, switches to build basic networks .	3
CO3	<u>Implement</u> various routing strategies on networks.	3
CO4	<u>Design</u> access control lists for traffic filtering.	6

Lab Details:

1. Study Ethernet, its types, specifications, UTP standards, and implement different cable constructions: straight-through, crossover, rollover.
2. Implement and understand basic network utilities: ping, ipconfig/ifconfig, arp, nslookup, tracert.
3. Implementation of access control protocols.
4. Implementation of flow control protocols.
5. Network traffic capture and analysis using Wireshark.
6. Study the devices found at different TCP/IP Model layers and their working.
7. Introduction to Cisco Packet Tracer: Basic CLI Modes, building a LAN with Hubs and Switches, understanding address learning in switches.
8. Learn end-to-end configuration of end devices: IP addressing (static and dynamic).
9. Configure routers using Packet Tracer and demonstrate subnetting.
10. Perform static and default routing on a network.
11. Implement RIP routing protocol (classful and classless addressing-based networks).
12. Implement EIGRP routing protocol (classful and classless addressing-based networks).
13. Configuring traffic filtering on routers via Standard access control list (ACL) configuration: blocking host and entire network.

14. Configuring traffic filtering on routers via Extended access control list (ACL)
configuration: block HTTP server.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
3 rd Year (6 th Semester)	Machine Learning Lab	ITL352	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: Computer Programming.

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

COs	Course Outcomes	Bloom's Taxonomy Level
C01	<u>Apply</u> machine learning algorithms to learn from data	3
C02	<u>Illustrate</u> the use of regularization techniques on the performance of the machine learning algorithms	4
C03	<u>Explore</u> the efficacy of neural networks on processing the images	4
C04	<u>Evaluate</u> various unsupervised learning to reduce the data with minimal loss of information	5

Lab Details:

1. Introduction

- a. Programs to understand the working of python

2. Machine Learning Libraries

- a. Programs to understand the use of numpy, Matplotlib and Pandas Library

3. Data Processing in Library

- a. Programs to understand the mechanism of importing datasets in python
- b. Programs to understand the processing of large datasets in python

4. Regression

- a. Programs to implement the gradient descent algorithm
- b. Programs to understand the use Linear Regression in learning the dataset relations

5. Logistic Regression

- a. Programs to understand the use of logistic regression
- b. Implement the logistic regression on the IRIS Dataset or on any other dataset

6. Regularization

- a. Implement the Ridge and Lasso Regularization
- b. Analyze the effect of regularization

7. Multiclass Classification

- a. Implementation of multiclass classification

- b. Softmax Regression
- 8. Naive Bayes**
 - a. Programs to implement naive bayes
 - b. Text classification implementation
- 9. Non Parametric Models**
 - a. Programs to implement KNN
 - b. Implement Support Vector Machines
- 10. Neural Networks**
 - a. Implement neural networks
 - b. Implement backpropagation
- 11. Neural Networks 2**
 - a. Implement image classification
- 12. Clustering**
 - a. Implement K-means clustering
 - b. Implement clustering on a real dataset
- 13. Principal Component Analysis**
 - a. Implement PCA
 - b. Implement PCA to compress the dataset
- 14. Reinforcement Learning**
 - a. Programs to understand the reinforcement learning

Semester VII

Year (Semester)	Course Title	Course Code	L-T-P-Credits
4 th Year (7 th Semester)	Image Processing	ITT401	2-1-0-3
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
	26 Marks	24 Marks	50 Marks

Pre-requisites: C/C++ programming, Signal and Systems, Introduction to Computer Graphics

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

CO No.	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Assess</u> what a digital image is, its acquisition and the steps involved in image processing	5
CO2	<u>Compare</u> various image enhancement techniques in spatial and frequency domain.	5
CO3	<u>Analyze</u> different causes of image degradation and image restoration. <u>Elucidate</u> the mathematical modeling of image. <u>Apply</u> colour image processing techniques, colour models.	4
CO4	<u>Discover</u> the need for image segmentation, morphological image processing and representation and description	3

Detailed Syllabus:

Module No.	Contents	Hours
Module 1	Introduction: What is digital image processing? The origins of digital image processing, Fundamental steps in digital image processing, components of an image processing system Image sensing and acquisition, Image sampling and quantization, basic relationships between pixels, linear and non-linear operations.	8
Module 2	Image Enhancement in Spatial domain: Gray level transformations, histogram processing, enhancement using arithmetic/logic operations, spatial filtering, smoothing and sharpening. Image enhancement in Frequency Domain: Fourier transform and frequency domain, smoothing and sharpening frequency domain filters	12
Module 3	Image Restoration: A Model of the Image Degradation/Restoration Process. Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering. Constrained Least Squares Filtering, Geometric Mean Filter, Geometric Transformations.	12

	Colour Image Processing: Fundamentals, models, colour transformations, smoothing and sharpening, colour segmentation and noise.	
Module 4	Image Segmentation: Detection of discontinuities, edge linking and boundary detection, thresholding, region based segmentation, morphological watersheds. Representation and description: Representation, boundary descriptors, regional descriptors, relational descriptors. morphological image processing.	10

Text Books:

1. Rafael C Gonzalez, Richard E Woods, Digital Image Processing - Pearson Education
2. Rafael C Gonzalez, Richard E Woods, Digital Image Processing with MATLAB- Pearson Education.

Reference Books:

1. William K Pratt, Digital Image Processing, John Willey
2. A.K. Jain, PHI, Fundamentals of Digital Image Processing, Pearson Education.
3. Chanda & Majumdar, "Digital Image Processing and Analysis", PHI.
4. Mark Nelson, Jean-Loup Gailly "The Data compression Book", bpb Publications.

Entrepreneurship Development (HST006)	Year & Semester: B. Tech 4th & 7th Sem		Total Course Credit: 3		
			L	T	P
			3	0	0
Evaluation Policy	Mid-Term (26 Marks)	Continuous Assessment (24 Marks)	End-Term (50 Marks)		

COURSE OUTCOMES:

After completing this course, the student must demonstrate the knowledge and ability to:

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Apply</u> the concepts of entrepreneurship and innovation.	3
CO2	<u>Analyze</u> entrepreneurship considering various theories and models.	4
CO3	<u>Appraise</u> Training Programme to inculcate Entrepreneurial acumen.	5
CO4	<u>Develop</u> Entrepreneurship Development Skills.	6

Detailed Syllabus:

Module-1	Entrepreneurship: Importance and growth, Characteristics of entrepreneurs, Ethical and social responsibilities of entrepreneurs, Challenges and opportunities of different forms of entrepreneurship, Entrepreneurial motivation.	10 Hrs
Module-2	Theories of entrepreneurship, Schumpeter's Theory of Innovation, Economic Theory of Entrepreneurship, Resource based, Opportunity based, psychological and Sociological theories of entrepreneurship.	10 Hrs
Module-3	Designing Appropriate Training Programme to inculcate Entrepreneurial Spirit, Training for New and Existing Entrepreneurs, Feedback and Performance of Trainees, Training entrepreneurs for creative problem solving.	12 Hrs
Module-4	Entrepreneurship Development Skills: Meaning of Entrepreneurship skill, Types of Entrepreneurship Skills: Business management skills, Teamwork and leadership skills, Time management and organizational skills	10 Hrs

Books Recommended:

Text books	1.	Kumar, A. (2012). <i>Entrepreneurship: Creating and Leading an Entrepreneurial Organization</i> . Pearson Education India.
	2.	Rao, T., & Kuratko, D. (2012). <i>Entrepreneurship</i> . Cengage learning India.
	3.	Ramachandran, K. (2012). <i>Entrepreneurship Development</i> . McGraw Hill Education India.
Reference books	4.	Roy, R. (2020). <i>Entrepreneurship (Ed. 3rd)</i> . Oxford University Press India.
	5.	Chole, R. R., Kapse, P. S., & Deshmukh, P. R. (2012). <i>Entrepreneurship Development and Communication Skills</i> . Scientific Publisher.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
4 th Year (7 th Semester)	Image Processing Lab	ITL403	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: C programming

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

COs	Course Outcomes	Bloom's Taxonomy Level
CO1	<u>Apply</u> programming for image-processing based problems and <u>use</u> various basic components of Matlab .	3
CO2	<u>Assess</u> and <u>implement</u> basic image processing techniques, diverse image enhancement techniques in spatial and frequency domain.	5
CO3	<u>Illustrate</u> colour image processing techniques and image restoration.	4
CO4	<u>Assess</u> and <u>implement</u> image segmentation techniques.	5

Lab Details:

Week 1	Basics of MATLAB 1. Installation of MATLAB 2. Introduction to MATLAB
Week 2	Matrix operations I 1. Enter 3 matrices of 4 ×4 size and display them. 2. Create matrices with <i>zeros</i> , <i>eye</i> and <i>ones</i> . 3. Array and matrix indexing.
Week 3	Matrix operations II 1. Create a big matrix using submatrices. 2. Manipulate the above matrix in the following ways: a. Delete/ display some rows/columns. b. Extract/replace some some submatrices. c. Create a symmetric matrix. 3. Learn to use <i>plot</i> and <i>fplot</i> functions.

Week 4	Basics of image processing I <ol style="list-style-type: none"> 1. Read and display an image. 2. Check how the image appears in the workspace. 3. Write the image to a disk file. 4. Learn usage of <i>whos</i>, <i>size</i>, <i>figure</i>, <i>imfinfo</i>, <i>disp</i>.
Week 5	Basics of image processing II <ol style="list-style-type: none"> 1. Conversion between data classes and types. (use any random matrix.) 2. Usage of arithmetic, relational and logic operators.
Week 6	Image enhancement I <ol style="list-style-type: none"> 1. Learn the usage of intensity transformation function: <i>imadjust</i>. 2. Use the Logarithmic and contrast stretching transformations on an image. Display resultant image.
Week 7	Image enhancement II <ol style="list-style-type: none"> 1. Generate and plot histogram of any random image. (use <i>imhist</i>, <i>bar</i>, <i>stem</i>, <i>plot</i>) 2. Implement image histogram equalisation. 3. Implement image histogram matching.
Week 8	Image enhancement III <ol style="list-style-type: none"> 1. Perform shrinking of an Image. 2. Perform Zooming of an Image. 3. Perform cropping of an Image.
Week 9	Image enhancement IV <ol style="list-style-type: none"> 1. Implement Low Pass Filter. Analyse results for ideal, Gaussian and Butterworth low/high pass filters. 2. Implement High Pass Filter. Analyse results for ideal, Gaussian and Butterworth low/high pass filters
Week 10	Image enhancement V <ol style="list-style-type: none"> 1. Implement linear spatial filtering. (<i>imfilter</i>, <i>fspecial</i>) 2. Use various options for function <i>imfilter</i>. 3. Implement nonlinear spatial filtering. (<i>colfilt</i>, <i>ordfilt2</i>, <i>medfilt2</i>).
Week 11	Image Restoration <ol style="list-style-type: none"> 1. To add noise use <i>imnoise</i> 2. Model a blurred and noisy image. 3. Implement inverse filtering (Restoration). Use all three methods.
Week 12	Colour Image Processing I <ol style="list-style-type: none"> 1. RGB to grayscale. 2. RGB to HSI and vice versa. 3. RGB to CMY and vice versa. 4. Grayscale to pseudo- color using equations.

Week 13	Colour Image Processing I <ol style="list-style-type: none">1. Implementing tone correction.2. Implement smoothing functions on an image.3. Implement sharpening functions on an image
Week 14	Image segmentation <ol style="list-style-type: none">1. Implement Edge Detection.2. Implement Line Detection.

Year (Semester)	Course Title	Course Code	L-T-P-Credits
4 th Year (7 th Semester)	Compiler Design Lab	ITL404	0-0-2-1
Evaluation Policy	Mid-Term	Internal Assessment	End-Term
100 Marks	40	-	60

Pre-requisites: None.

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

COs	Course Outcomes	Bloom's Taxonomy Level
C01	<u>Demonstrate</u> use of lex and yacc tools for developing a scanner and a parser.	3
C02	<u>Design</u> and <u>implement</u> LL and LR parsers.	6
C03	<u>Use</u> different tools in construction of the phases of a compiler for the mini language.	3
C04	<u>Design</u> , <u>develop</u> , and <u>implement</u> a compiler for any language.	6

Lab Details:

1. Basic

- a. Write a program to create a functions for string handling.
i) Strlen(); ii) Strcpy(); iii) Strcat(); iv) Strcmp(); v) Strupr();
- b. Write a C program to Scan and Count the number of characters, words, and lines in a file.

2. Lexical Analyzer

- a. Design a lexical analyzer for the given language. The lexical analyzer should ignore redundant spaces, tabs and new lines, comments etc.
- b. Implementation of Lexical Analyzer using Lex Tool.

3. Tokens, Lexemes and Patterns

- a. Write a program to identify whether a given line is a comment or not.
- b. Write a program to check whether a given identifier is valid or not.
- c. Write a program to recognize strings under 'a', 'a*b+', 'abb'.

4. Lexical Analyzer tools

- a. Implement the lexical analyzer using JLex and flex tool.
- b. Study the YACC tool and Evaluate an arithmetic expression with parentheses, unary and binary operators using Flex and Yacc (CALCULATOR)

5. NFA'S & DFA'S

- a. Write a C program to implement NFAs that recognize identifiers, constants, and operators of the mini language.
- b. Write a C program to implement DFAs that recognize identifiers, constants, and operators of the mini language. \
- c. Write a program to minimize any given DFA.
- d. Write a program to find ϵ – closure of all states of any given NFA with ϵ transition.

6. Study of YACC and JFLAP

- a. Generate YACC specifications for a few syntactic categories.
 - i) Program to recognize a valid arithmetic expression that uses operator +, −, * and /.
 - ii) Program to recognize a valid variable which starts with a letter followed by any number of letters or digits.
 - iii) Using JFLAP, create a DFA from a given regular expression.

7. Semantic Analyzer and Symbol Table

- a. Write a C program to implement Program semantic rules to calculate the expression that takes an expression with digits, + and * and computes the value.
- b. To write a C program to implement a symbol table.

8. Top Down Pasing

- a. Write a program to implement Predictive Parsing algorithm.

9. LL(1) Parsing

- a. Create LL(1) parse table for a given CFG and hence Simulate LL(1) Parsing
- b. Write functions to find FIRST and FOLLOW of all the variables.

10. Bottom Up Parsing

- a. Write a program to design a LALR bottom up parser for the given language.
- b. Using JFLAP create SLR(1) parse table for a given grammar. Simulate parsing and output the parse tree proper format.
- c. Develop an operator precedence parser for a given language.

11. Syntax Directed Translation

- a. Write program to generate machine code from the abstract syntax tree generated by the parser. The following instruction set may be considered as target code.
- b. Convert the BNF rules into YACC form and write code to generate abstract syntax tree.

12. Code Generation

- a. Write a C program to generate three address code.
- b. Implement the back end of the compiler which takes the three address code and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be simple move, add, sub, jump etc.

13. Run time Environment

- a. Write a program to perform loop unrolling.
- b. Write a program to perform constant propagation.

14. Case Study

- a. UML Diagram for ATM Transaction System.
- b. UML Diagram for College Administration System.

Semester VIII