UNIT - I (13 Periods)


UNIT - II (13 Periods)


UNIT - III (12 Periods)


UNIT - IV (12 Periods)


TEXT BOOK:

REFERENCE BOOKS:
UNIT – I  (16 Periods)

**Foundations**: Sets, Relations and Functions, Fundamentals of Logic, Logical Inferences, Methods of Proof of an implication, First order Logic & Other methods of proof, Rules of Inference for Quantified propositions, Mathematical Induction.

UNIT – II  (15 Periods)

**Elementary Combinatorics**: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with repetitions, Enumerating Permutation with Constrained repetitions.

**Recurrence relations**: Generating functions of sequences, Calculating Coefficients of Generating Functions.

UNIT – III  (15 Periods)

**Recurrence Relations**: Solving recurrence relations by Substitution and generating functions. The methods of characteristic roots, solutions of inhomogeneous recurrence relations.

**Relations and digraphs**: Special properties of binary relations, Operations on relation.

UNIT – IV  (14 Periods)

Ordering relations, Lattice, Paths and Closures, Directed Graphs and Adjacency Matrices, Application: Topological Sorting.

**Graphs**: Basic Concepts, Isomorphisms and Subgraphs, Planar Graphs, Euler's Formula; Multigraphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four Color Problem.

**TEXT BOOK**:  

**REFERENCE BOOKS**:  
1. C.L. Liu, "Elements of Discrete Mathematics".  
UNIT – I (13 Periods)
Review of Number systems & codes, Representation of integers and Floating point numbers, Accuracy, Introduction to integer arithmetic operations.

BOOLEAN ALGEBRA AND LOGIC GATES: Basic Definitions, Axiomatic definition of Boolean Algebra, Basic theorems and Properties of Boolean Algebra, Boolean functions, Canonical and Standard Forms, Other operations, Digital Logic Gates.

SIMPLIFICATION OF BOOLEAN FUNCTIONS: The Map Method, Two and three variable Maps, Four-variable Map, Five and six-variable Maps, Product of Sums Simplification, NAND and NOR implementation, Don't-Care conditions.

UNIT – II (13Periods)
SIMPLIFICATION OF BOOLEAN FUNCTIONS: The Tabulation Method, Determination of Prime Implicants, Selection of Prime-Implicants.


COMBINATIONAL LOGIC WITH MSI AND LSI: Binary parallel adder, Decimal adder, Magnitude comparator, Decoders, Multiplexers.

UNIT – III (12 Periods)

UNIT – IV (12 Periods)
REGISTERS, COUNTERS: Registers, Shift registers, Ripple counters, Synchronous counters, Timing sequences.

MEMORIES: Classification of ROMs, EPROMs, EEPROMs, RAMs.

PROGRAMMABLE LOGIC: Read only memory (ROM), Programmable logic device (PLD),programmable logic array (PLA), Programmable array logic (PAL).

TEXT BOOK:
1. Morris Mano, "Computer Engineering Hardware Design", PHI.

REFERENCE BOOKS:
2. Donald e Givone, "Digital Principles and Design", TMH.
UNIT – I                                        (16 Periods)
System Structures: OS Services, System Calls, Types of System Calls, System Programs, OS Design and Implementation, OS Structure.

UNIT – II                                       (15 Periods)
Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, //Thread Scheduling, //Multiple-Processor Scheduling, OS Examples, Algorithm Evaluation.
Synchronization: Background, Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic problems of Synchronization, Monitors.

UNIT – III                                      (15 Periods)
Memory-Management Strategies: Background, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation.
Virtual-Memory Management: Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Other Considerations.

UNIT – IV                                      (14 Periods)

TEXT BOOK:

REFERENCE BOOKS:
DATA STRUCTURES
14CS305
B.Tech.,(Semester- III)

<table>
<thead>
<tr>
<th>Lectures</th>
<th>4 Periods/Week, Tutorial: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Exam</td>
<td>3 hours</td>
</tr>
<tr>
<td>Continuous Assessment</td>
<td>40</td>
</tr>
<tr>
<td>Final Exam Marks</td>
<td>60</td>
</tr>
</tbody>
</table>

UNIT – I (16 Periods)

Algorithm Analysis: Mathematical Background, Model, What to Analyze, Running Time Calculations.


UNIT – II (15 Periods)

Stacks and Queues: The Stack ADT and its applications such as Infix to Postfix expression conversions, Evaluation of Postfix expressions, Delimiter Matching. The Queue ADT, The Circular Queue ADT.

Sorting Preliminaries: Shellsort, Mergesort, Quicksort.

UNIT – III (15 Periods)


UNIT – IV (14 Periods)

Hashing: General Idea, Hash Function, Separate Chaining, Open Addressing, Linear Probing, Priority Queues (Heaps), Model, Simple implementations, Binary Heap, Heap Sort.

Graphs: Definitions, Representations: Adjacency matrices and lists, Graph traversals: Depth first, Breadth first.

TEXT BOOK:

REFERENCE BOOKS:
UNIT – I  
The Creation of C#
An Overview of C#
Data Types, Literals, and Variables
Operators
Program Control Statements
Introducing Classes and Objects
Arrays and Strings
A Closer Look at Methods and Classes

UNIT – II  
Operator Overloading
Indexers and Properties: Properties, Use Access Modifiers with Accessors
Inheritance
Interfaces, Structures, and Enumerations

UNIT – III  
Exception Handling
Delegates, Events-Delegates, Anonymous Functions, Anonymous Methods, Events.

UNIT – IV  
Namespaces, the Preprocessor, and Assemblies

Text Book:
Reference Books:
1. Programming C# 5.0 by Ian Griffiths, O'REILLY, 2012.
LIST OF EXPERIMENTS

1. **BODY LANGUAGE**
   a. Facial Expressions.
   b. Kinesics.
   c. Oculesics.
   d. Haptics.
   e. Proxemics.
   f. Para Linguistics.

2. **LIFE SKILLS**
   a. Positive Attitude
   b. Social Behavior & Social Norms.
   c. Ethics, Values and Positive Work Ethics.
   d. Time Management

3. **EMOTIONAL INTELLIGENCE**
   a. Self Awareness through Johari Window and SWOT analysis.
   b. Self Control.
   c. Self Motivation.
   d. Empathy.
   e. Social Skills.
   f. Self Esteem.
   g. Managing stress.
   h. Assertiveness.

4. **PROBLEM SOLVING SKILLS**
   a. Critical Thinking and Brain Storming
   b. Lateral Thinking and Six Thinking Hats.
   c. Creative Thinking.
   d. Conflict Management.

5. **EMPLOYABILITY SKILLS**
   a. Group Discussion.
   b. Team Building and Leadership Qualities
   c. Interview Skills.
REFERENCE BOOKS:

6. "The 7 Habits Of Highly Effective People", Stephen R. Covey
LIST OF EXPERIMENTS

1. Code the following list ADT operations using array, single linked list, double linked list.
   (a) void is_emptyList(List L)
   (b) List makeNullList(size n)
   (c) Position firstPost(List L)
   (d) Position endPost(List L)
   (e) Position nextPost(List L, Position p)
   (f) Position prevPos(List L, position p)
   (g) Position find(List L, Element x)
   (h) Position findKth(List L, int k)
   (i) void insert(List L, Position p)
   (j) void delete(List L, Position p)
   (k) void append(List L, Element x)
   (l) int cmp(List L, Position p1, Position p2)
   (m) int cmp2(List L, List L, Position p1, Position p2)
   (n) void swap(List L, Position p1, Position p2)
   (o) Element retrieveElement(List L, Position p)
   (p) void printElement(List L, Position p)

2. Using the above List ADT operations, Write a menu driven program to support following higher level list operations:
   (a) Create null list
   (b) Read a list of elements into the list.
   (c) Insert an element in the Kth position of the list
   (d) Delete an element in the Kth position of the list
   (e) Delete a given element from the list
   (f) Find whether given element is present in the list
   (g) Display the elements of the list

3. Write a program that reads two lists of elements, prints them, reverses them, prints the reverse list, sort the lists, print the sorted lists, merges the list, prints merge list.

4. Implement a polynomial ADT and write a program to read two polynomials and print them, adds the polynomials, prints the sum, multiply the polynomials and print the product.

5. Implement stack ADT and write a program that reads an infix arithmetic expression
of variables, constants, operators (+, -, *, /) and converts it into the corresponding postfix form. Extend the program to handle parenthesized expression also.

6. Implement Queue ADT and write a program that performs Radix sort on a given set of elements.

7. Implement the following sorting operations:-
   (a) Shell Sort
   (b) (b) Heap Sort
   (c) (c) Merge Sort
   (d) (d) Quick Sort

8. Implement Binary Tree ADT and write a program that reads postfix Arithmetic expression form, builds the expression tree and performs tree Traversal on it.

9. Implement Binary search ADT and write a program that interactively allows
   (a) Insertion   (b) Deletion   (c) Find_min  (d) Find_max  (e) Find operations

10. Implement AVL Tree ADT and Write a program that interactively allows
    (a) Insertion   (b) Deletion   (c) Find_min  (d) Find_max
    Implement Hashing and Write a program to find a element using Open Addressing.
LIST OF EXPERIMENTS

1. Implement a class List and the list operations. Use all possible basic features of C#.
2. Write a C# program to demonstrate Arrays (2-D and jagged).
3. Design a class to demonstrate String class methods.
4. Design an appropriate class that represents a mathematical entity and provide the operations with Operator Overloading.
5. Implement a class hierarchy with Abstract Classes, Virtual methods & Overriding.
6. Implement a class clock that publishes seconds change event. Design classes that subscribe to the event with respective behaviours.
7. Design a Data Structure with Exception Handling.
8. Write a program to demonstrate Generic Class Generic Method.
9. Write a program to demonstrate Collections and Generic Collections.
10. Write a C# program to determine the Generic Classes Generic Methods and Generic Interfaces.