Object Oriented Programming through C++

UNIT-I:
Objectives: Exposure to basics of object oriented mode, C++ programming and I/O in C++


INPUT AND OUTPUT IN C++:
Introduction, Streams in C++ and Stream classes, Pre-Defined Streams, Stream Classes, Formatted and Unformatted Data, Unformatted Console I/O Operations, Member Functions of Istream class, formatted Console I/O Operations, Bit Fields, Flags without Bit Field, manipulators, user defined manipulators.

UNIT-II:
Objectives: Focus on Basic Concepts in C++ programming, Operators, control structures, functions, overloading, recursion

Tokens in C++, Variable Declaration and Initialization, Data Types, Operators in C and C++, Scope Access Operator, NameSpace, Memory Management Operators, Comma Operator, Revision of Decision Statements, Control Loop statements.

FUNCTIONS IN C++: Introduction, Structure of Function, Passing Arguments, Lvalues and Rvalues, Return by Reference, Returning more values by Reference, Default Arguments, Const Arguments, Inputting Default Arguments, Inline Functions, Function Overloading, Principles of Function Overloading, Recursion.

UNIT-III:
Objectives: Acquaintance with classes, objects and member functions

CLASSES AND OBJECTS: Introduction, Classes in C++, declaring Objects, Access Specifiers and their scope, Member Functions, Outside member function as Inline, Data Hiding and Encapsulation, Classes, Objects and Memory, Static Member variables, Static Member Functions, Static Object, Array of Objects, Objects as Function Arguments, Friend Functions, The Const Member Function, The Volatile Member Function, Recursive Member Function, Local classes, Empty, Static and Const Classes, Member Function and Non-Member Function, Overloading Member Functions, Nested Class.

UNIT-IV:
Objectives: Focus on constructors, destructors, variants in them, operator overloading, type conversions

CONSTRUCTORS AND DESTRUCTORS: Introduction, Characteristics of Constructors & Destructors, Applications with constructors, parameterized Constructor, Overloading Constructors(Multiple Constructors), Array of Objects using Constructors, Constructors with Default Arguments, Copy Constructors, The Const Objects, Destructors, Calling Constructors and Destructors, Qualifier and Nested Classes, Anonymous Objects, Private Constructors and Destructors, Dynamic Initialization using Constructors, Dynamic Operators and Constructors, Recursive Constructor, Constructor and Destructor with Static Members, Local vs Global Object.

UNIT-V:
Objectives: Concentration on Inheritance, types of Inheritance, polymorphism, virtual functions.

INHERITANCE: Introduction, Reusability, Access Specifiers and Simple Inheritance, Protected Data with Private Inheritance, Types of Inheritances(Single Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance, Multipath Inheritance), Virtual Base Classes, Constructors, Destructors and Inheritance, Object as a Class member, Abstract classes, Qualifier Classes and Inheritance, Constructors in Derived class, Pointers and Inheritance, Overloading Member function, Advantages of Inheritance, Disadvantages of Inheritance.

BINDING, POLYMORPHISM AND VIRTUAL FUNCTIONS: Introduction, Binding in C++, Static (Early) Binding, Dynamic(late) Binding, Pointer to Base and Derived class objects, Virtual Functions, Rules for Virtual Functions, Array of Pointers, Pure Virtual Functions, Abstract classes, Working of Virtual Functions, Virtual Functions in Derived classes, Object Slicing, Constructors and Virtual Functions, Virtual Destructors, Destructor and Virtual Functions.

UNIT-VI:
Objectives: Focus on Files, File Operations, generic programming, templates, function templates, Exception handling


GENERIC PROGRAMMING WITH TEMPLATES: Introduction, Need of Template, Definition of class Template, Normal function Template, Working of Function Template, Class template with more parameters, Functions Templates with more arguments, Overloading of Template Functions, member function templates, Recursion with template function, class template with Overloaded operators, class template revisited, class templates and Inheritance, Container Classes, Types of Containers, Container Adaptors, Iterators.

EXCEPTION HANDLING: Introduction, Principles of Exception Handling, the keywords, try, throw and catch, Exception Handling Mechanism, Multiple Catch Statements, Catching Multiple Exceptions, Re-Throwing Exception, Specifying Exception, Exceptions in Constructor and Destructors, Controlling Uncaught Exceptions, Class template with exception handling

TEXTBOOKS:
4. Object Oriented Programming with C++, 2nd ed, Sourav Sahay, OXFORD.

REFERENCE BOOKS:
1. The Complete Reference, C++, 4th ed, Herbert Schildt, TMH
MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

Objectives: Acquaintance with the basic mathematical implication for computer science, applications of mathematics in computer science

UNIT: I

Objective: Acquiring the relevance of statements, inferences and predicates in computer science

Mathematical Logic:


Predicate calculus: Predicative Logic, Statement Functions, Variables and Quantifiers, Free & Bound Variables, Inference theory for predicate calculus

UNIT-II

Objective: Overview of Number theory, basic algorithms in number theory and mathematical induction

Number of Theory & Induction:

Properties of integers, Division Theorem The Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple Testing for Prime Numbers The Fundamental Theorem of Arithmetic Modular Arithmetic (Fermat’s Theorem and Euler’s Theorem)

Mathematical Induction: Principle of Mathematical Induction, exercises

UNIT III:

Objective: Focuses on sets and relations and their operations, relations and functions

Set Theory:


Functions: Bijective Functions Composition of Functions, Inverse Functions, Permutation Functions, Recursive Functions

UNIT IV:

Objective: Exposure of graphs, their representation, types, trees and tree variants

Graph Theory:

Basic Concepts of Graphs, Sub graphs, Matrix Representation of Graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian Graphs, Hamiltonian Graphs, Multigraphs, (Problems and Theorems without proofs)

Planar Graphs, Euler’s Formula, Graph Colouring and Covering, Chromatic Number (Problems and Theorems without proofs)


UNIT V:

Objective: Overview of algebraic structures, Group theory, Binomial theorem, permutations and combinations
Algebraic structures: Lattice: Properties, Lattices as Algebraic Systems, Algebraic Systems with one Binary Operation Properties of Binary operations Semi groups and Monoids: Homomorphism of Semi groups Monoids Groups: Abelian Group Subgroups Cosets (Definitions and Examples of all structures) Algebraic Systems with two Binary Operations: Rings


Binomial Theorem: Binomial and Multinomial Coefficients, Generating Functions of Permutations and Combinations Principles of Inclusion – Exclusion Problems

UNIT VI

Objective: Overview of generating functions, recurrence relations and solving recurrence relations

Recurrence Relation:

Generating Function of Sequences Partial Fractions Calculating Coefficient of Generating Functions

Recurrence Relations Formulation as Recurrence Relations Solving linear homogeneous recurrence Relations by substitution, Generating functions and The Method of Characteristic Roots.

Solving I homogeneous Recurrence Relations

TEXT BOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science, Tremblay, Manohar, TMH
2. Discrete Mathematics for Computer Scientists & Mathematicians, 2/e, Mott, Kandle, Baker, PHI
3. Discrete Mathematics, Swapan Kumar Chakraborthy, Bikash kanti sarkar, OXFORD
4. Discrete mathematics and Graph theory, 3rd ed, Biswal, PHI

REFERENCE BOOKS:

2. Discrete Mathematics, S.Santha, Cengage
3. Discrete Mathematics with Applications, Thomas Koshy, Elsevier
4. Discrete Mathematics, 2/e, JK HSarma, Macmillan
Unit I: (*The Learning objective of this Unit is to understand the concepts and nature of Managerial Economics and its relationship with other disciplines, Concept of Demand and Demand forecasting)


(**The Learner is equipped with the knowledge of estimating the demand for a product and the relationship between Price and Demand)

Unit-II: (*The Learning objective of this Unit is to understand the concept of Production function, Input Output relationship, different Cost Concepts and Concept of Cost-Volume-Profit Analysis)

Production and Cost Analyses:

(** One should understand the Cost Concepts for decision making and to estimate the least cost combination of inputs).

Unit-III: (*The Learning Objective of this Unit is t understand the Nature of Competition, Characteristics of Pricing in the different Market Structure and significance of various pricing methods)


(***One has to understand the nature of Different markets and Price Output determination under various market conditions)

UNIT-IV :(* The Learning objective of this Unit is to know the different forms of Business organization and their Merits and Demerits both public & private enterprises and the concepts of Business Cycles:

(*** One should equipped with the knowledge of different Business Units)

Unit V: (*The Learning Objective of this Unit is to understand the different Accounting Systems preparation of Financial Statement and uses of different tools for performance evaluation)

Introduction to Accounting & Financing Analysis:
Introduction to Double Entry Systems – Preparation of Financial Statements – Analysis and Interpretation of Financial Statement – Ratio Analysis – Preparation of Funds flow cash flow statements (Simple Problems)

(*** The Learner is able to prepare Financial Statements and the usage of various Accounting tools for Analysis)
Unit VI (*The Learning objective of this Unit is to understand the concept of Capital, capitalization, Capital budgeting and to know the techniques used to evaluate Capital Budgeting proposals by using different methods*)

**Capital and Capital Budgeting:** Capital Budgeting: Meaning Capital-capitalization-meaning of Capital Budgeting-Traditional and Modern Methods.

(**The Learner is able to evaluate various investment project proposals with the help of capital budgeting techniques for decision making)**

Note: *Learning Objective

**Learning Assessment

Text Books:


References:

1. V. Maheswari: Managerial Economics, Sultan Chand.
2. Suma Damodaran- Managerial Economics, Oxford 2011
DATA STRUCTURES

Objectives: Comprehensive knowledge of data structures and ability to implement the same in software applications

UNIT I:
Objectives: exposure to algorithmic complexities, recursive algorithms, searching and sorting techniques
Recursion: Preliminaries of Algorithm, Algorithm analysis and complexity, Data structure- Definition, types of data structures
Recursion: Definition, Design Methodology and Implementation of recursive algorithms, Linear and binary recursion recursive algorithms for factorial function, GCD computation, Fibonacci sequence, Towers of Hanoi, Tail recursion
List Searches using Linear Search, Binary Search, Fibonacci Search
Sorting Techniques: Basic concepts, sorting by: Insertion (Insertion sort), Selection (heap sort), exchange (bubble sort, quick sort), distribution (radix sort) and merging (merge sort) Algorithms.

UNIT II:
Objectives: Applying stack and queue techniques for logical operations Stacks and Queues:
Basic Stack Operations, Representation of a Stack using Arrays, Stack Applications: Reversing list, Factorial calculation, In-fix- to postfix Transformation, Evaluating Arithmetic Expressions.
Queues: Basic Queue Operations, Representation of a Queue using array, Implementation of Queue Operations using Stack, Applications of Queues- Round Robin Algorithm, Circular Queues, Priority Queues

UNIT- III:
Objectives: Exposure to list representation models in various types of applications
Linked Lists: Introduction, Single linked list, Representation of a linked list in memory, Operations on a single linked list, Merging two single linked lists into one list, Reversing a single linked list, Applications of single linked list to representing polynomial expressions and sparse matrix manipulation, Advantages and disadvantages of single linked list, Circular linked list, Double linked list

UNIT-V:
Objectives: Advanced understanding of other variants of trees and their operations
Advanced concepts of Trees: Tree Travels using stack (non recursive), Threaded Binary Trees. Binary search tree, Binary search tree- Basic concepts BST operations: insertion, deletion Balanced Binary trees-need, basics and applications in computer science (No operations)

UNIT-VI:
Objectives: orientation on graph, representation of graphs, graph traversals, spanning trees
Graphs: Basic Concepts, Representation of Graphs: using Linked list and adjacency matrix, Graph algorithms
Graphs Traversals (BFS & DFS) Applications: Dijkstra’s shortest path, Transitive closure Minimum Spanning Tree using Prim’s Algorithm, Warshall’s Algorithm (Algorithmic Concepts Only, no Programs Required).

TEXT BOOKS:
1. Data Structure with C, Seymour Lipschutz TMH
2. Data Structures using C, Reema Thareja, Oxford
3. Data Structures, 2/e, Richard F, Gilberg, Forouzan, Cengage

REFERENCES BOOKS:
2. Classic Data Structures, 2/c, Debasis, Samanta, PHI, 2009
DIGITAL LOGIC DESIGN

UNIT I: Number Systems
Binary, Octal, Decimal, Hexadecimal Number System. Conversion of Numbers from One Radix to another Radix r’s complement and (r-1)’s complement subtraction of unsigned numbers problems Signed binary numbers Weighted and non weighted codes

UNIT II: Logic gates And Boolean Algebra
Basic Gates NOT, AND, OR, Boolean theorems, Complement and Dual of Logic Expressions, Universal Gates, XOR and XNOR Gates, SOP,POS Minimizations of Logic Functions Using Boolean Theorems, Two level Realization of Logic Functions Using Universal Gates.
**Gate-Level Minimization:** Karnaugh Map Method (K-Map): Minimization of Boolean functions maximum upto Four variables, POS and SOP, Simplifications with Don’t Care Conditions Using K-Map

UNIT III: Combinational Logic Circuits

UNIT IV: Introduction to Sequential Logic Circuits

UNIT V: Registers And Counters
Design of Registers, Buffer Register, Control Buffer Registers, Bidirectional Shift Registers, Universal Shift Register, Design of Ripple Counters, Synchronous Counters and Variable Modulus Counters, Ring Counter, Johnson Counter.

UNIT VI: Introduction to Programmable Logic Devices (PLDs)
PLA, PAS, PROM. Realization of Switching Functions Using PROM, PAL and PLA. Comparison of PLA, PAL and PROM.

TEXT BOOKS:
1. Digital Design, 4/e M.Morris Mano, Michael D Ciletti, PEARSON
2. Fundamentals of Logic Design, 5/e, Roth, Cengage

REFERENCE BOOKS
2. Digital Logic Design, Leach, Malvino, Saha, TMH
3. Modern Digital Electronics, R.P. Jain, TMH
DATA STRUCTURES LAB

Exercise 1:
Write recursive programme which computes the $n^{th}$ Fibonacci number, for appropriate values of $n$.
Analyze behavior of the programme Obtain the frequency count of the statement for various values of $n$.

Exercise 2:
Write recursive programme for the following
a) Write recursive C programe for calculation of Factorial of an integer
b) Write recursive C programe for calculation of GCD (n, m)
c) Write recursive C programe for Towers of Hanoi : N disks are to be transferred from peg S to peg D with Peg I as the intermediate peg.

Exercise 3:
a) Write C programs that use both recursive and non recursive functions to perform Linear search for a Key value in a given list.
b) Write C programs that use both recursive and non recursive functions to perform Binary search for a Key value in a given list.
c) Write C programs that use both recursive and non recursive functions to perform Fibonacci search for a Key value in a given list.

Exercise 4:
a) Write C programs that implement Bubble sort, to sort a given list of integers in ascending order
b) Write C programs that implement Quick sort, to sort a given list of integers in ascending order
c) Write C programs that implement Insertion sort, to sort a given list of integers in ascending order

Exercise 5:
a) Write C programs that implement heap sort, to sort a given list of integers in ascending order
d) Write C programs that implement radix sort, to sort a given list of integers in ascending order
e) Write C programs that implement merge sort, to sort a given list of integers in ascending order

Exercise 6:
a) Write C programs that implement stack (its operations) using arrays
b) Write C programs that implement stack (its operations) using Linked list

Exercise 7:
a) Write a C program that uses Stack operations to Convert infix expression into postfix expression
b) Write C programs that implement Queue (its operations) using arrays.
c) Write C programs that implement Queue (its operations) using linked lists

Exercise 8:
a) Write a C program that uses functions to create a singly linked list
b) Write a C program that uses functions to perform insertion operation on a singly linked list
c) Write a C program that uses functions to perform deletion operation on a singly linked list

Exercise 9:
d) Adding two large integers which are represented in linked list fashion.
e) Write a C programme to reverse elements of a single linked list.
f) Write a C programme to store a polynomial expression in memory using linked list
g) Write a C programme to representation the given Sparse matrix using arrays.
h) Write a C programme to representation the given Sparse matrix using linked list

Exercise 10:
a) Write a C program to Create a Binary Tree of integers
b) Write a recursive C program, for Traversing a binary tree in preorder, inorder and postorder.
c) Write a non recursive C program, for Traversing a binary tree in preorder, inorder and postorder.
d) Program to check balance property of a tree.

Exercise 11:
a) Write a C program to Create a BST
b) Write a C programme to insert a note into a BST.
c) Write a C programme to delete a note from a BST.
DIGITAL LOGIC DESIGN LAB

List of Experiments:

1. Verification of Basic Logic Gates

2. Implementing all individual gates with Universal Gates NAND & NOR.

3. Design a circuit for the given canonical form, draw the circuit diagram and verify the De-Morgan laws.

4. Design a Combinational Logic circuit for 4x1 MUX and verify the truth table.

5. Design a Combinational Logic circuit for 1x 4D – MUX and verify the truth table.

6. Verify the data read and data write operations for the IC 74189.

7. Design a Gray code encoder and interface it to SRAM IC 74189 for write operation display on 7-segment.

8. Design a Gray code De-Coder and interface it to SRAM IC 74189 for read operation display it on 7-segment.


10. Verification of truth tables of the basic Flip-Flops with Synchronous and Asynchronous modes.


12. Design a Decade Counter and verify the truth table.

13. Design the Mod 6 counter using D-Flip-Flop.

14. Construct 4-bit ring counter with T-Flip-Flop and verify the truth table.

15. Design a 8 – bit right shift Register using D-Flip – Flop and verify the truth table.
OBJECT ORIENTED PROGRAMMING LAB

1. Write a C++ program illustrating Variable Scope.
2. Write a C++ program illustrating Swapping integer values by reference.
3. Write a C++ program illustrating Checking whether the number is even or odd using Ternary operator.
4. Write a C++ program illustrating a program to find the roots of a quadratic equation. Use switch statements to handle different values of the discriminant \( b^2 - 4ac \)
5. Write a C++ program illustrating interactive program to multiply 2 variables after checking the compatibility.
6. Write a C++ program illustrating interactive program for computing the roots of a quadratic equation by handling all possible cases. Use streams to perform I/O operating.
7. Write C++ program illustrating sorting of integer number.
8. Write a C++ program illustrating factorial using recursion.
9. Write a C++ program illustrating pass by value, pass by address.
10. Write a C++ program illustrating Function Overloading.
11. Write a C++ program illustrating an interactive program for swapping integer, real, and character type variables without using function overloading. Write the same program by using function overloading features and compare the same with its C counterpart.
12. Write a C++ program illustrating inline functions.
13. Write a C++ program illustrating Friend function.
14. Write a C++ program illustrating Exception handling.
15. Write a C++ program illustrating function template.
16. Write a C++ program illustrating Overloading increment, decrement, binary+ & operator
17. Write a C++ program illustrating Virtual function.
18. Write a C++ program illustrating an interactive program to process complex numbers. It has to Perform addition, multiplication, and division of complex number. Print results in x+iy form. Create a class for the complex number representation.
19. Write a C++ program illustrating user defined string processing functions using pointers (string length, string copy, string concatenation)
20. Write a C++ program illustrating Constructor overloading (Both parametesed and default).
21. Write a C++ program illustrating copy constructor.
22. Write a C++ program illustrating access data members & member functions using ‘THIS’ pointer.
23. Write a C++ program illustrating overloading ++ operator to increment data.
24. Write a C++ program illustrating overloading of new and delete operator.
25. Write a C++ program illustrating Abstract classes.
26. Write a C++ program illustrating Inheritance (Multiple, Multilevel, Hybrid).
27. Write a C++ program illustrating virtual classes & virtual functions.
28. Write a C++ program illustrating overloading function template.
29. Write a C++ program illustrating class template.