

Content Product

Detailed syllabus

OBJECT ORIENTED PROGRAMMING AND DATA STRUCTURES

UNIT I - DATA ABSTRACTION AND OVERLOADING

Why Object?- Why C++?, Structure of C++ programs, Object oriented programming, Appreciate the evolution of OOPs, Features of Object Oriented Languages, Applications of OOPS, History of C++, Difference between Procedure Oriented Programming (POP) and Object Oriented Programming (OOP), History of C++, Features of C++, C++ Input and Output operators, Standard output (cout), Manipulator, Escape sequence, Standard input (cin). **Class and object creation** - Introduction, Class and object, Class definition, Class specification, Creation of objects, Access specifiers, Creating an objects, Pointers and objects, Constant objects, Objects as arguments, Arrays within a class, Arrays as objects, Nested classes. **Constructors and destructors** - Class constructors, Constructors, Types of constructors, Copy constructors, Overloaded constructors, Dynamic constructors, Explicit constructors, Destructor. **Member functions** - Function and data members, Class definition, Default arguments, Making an outside function inline, Nesting of member functions, Returning values from member function, Pointers to members. **Friend function** - Friend function, Inline functions, Advantages. **Dynamic memory allocation** - Static memory allocation, Dynamic memory allocation, new operator, Delete operator, Program to illustrate deleting of objects using delete operators. **Static Class Members** - Introduction to Static class member, Static Data Member, Static Member Function. **Iterators and containers** - Introduction. Components of STL, Containers, Algorithms, Iterators, Application of Container Class. **Proxy classes** - Proxy classes. **Operator overloading and function overloading** - Introduction to operator overloading, Program to return objects from function, Process and rules of operator overloading, Overloading unary operator, Overloading binary operator, Overloading binary operator using friend function, Overloading assignment operator, Manipulating strings using operators, Program on relational operator overloading, Function overloading.

UNIT II - INHERITANCE AND POLYMORPHISM

Necessity for inheritance, relation between base class and derived class- Introduction to inheritance, Inheritance terminologies, Access control over the members, Access rights of derived class, Relation between base class and derived class, Advantages of inheritance, Defining a class hierarchy. **Protected members** - Introduction to protected members. **Casting Class pointers** - Introduction to type casting, Casting operators, Overriding. **Visibility modes (access control)** - Introduction, Public derivation, Private derivation, Protected derivation, Making the private member of base class inheritable. **Types of inheritance** - Types of inheritance, Advantages of inheritance, Single inheritance, Multiple inheritance, Multilevel inheritance, Hierarchical inheritance, Hybrid inheritance. **Constructor and destructor in derived class** - Constructor in Derived Class, Constructor Rules for Derived Class, Constructor and destructor in derived class, Implicit derived class object to base class object conversion, Composition vs inheritance. **Pointers** - Pointers, Creating pointers to variables, Creating pointers to an array, Creating pointers to functions, Creating pointers to an objects, Pointers to objects, Program to illustrate pointers to objects, Program to illustrate array of pointers to objects, this pointer, Benefits of 'this' pointer, Example program of this pointer. **Abstract base classes and Concrete classes** - Abstract Class, Example Program for Abstract Class, Concrete class. **Polymorphism** - Polymorphism, Introduction to

Detailed syllabus

virtual function, What are virtual function?, Need for virtual function, Properties of virtual function, Declaration of virtual function, Example program of virtual function, Early binding or static binding, Late binding or dynamic binding, Pure virtual function, Rules for virtual class. **Virtual destructors** - Introduction to virtual destructors, Example of without and with virtual destructor, Pure virtual destructor.

UNIT III - LINEAR DATA STRUCTURES

Abstract Data Types (ADTs) – List ADT - Introduction to data structure, Representation of data structure, Implementation of data structure, Abstract Data Types (ADT), The list ADT, The array list ADT, Array based implementation, Growable array based array list. **Linked list implementation** - Introduction to linked list, Representation of linked list, Implementation using array, Linked list using dynamic variables, Linked implementation of Stacks, Advantages and Disadvantages of Linked list, Singly linked list, Operations on singly linked list, Advantages and Disadvantages, Circular linked list, Operations in circular linked list, Advantages and disadvantages of Circular linked list, Doubly linked list, Operations in doubly linked list. **Applications of lists** - Applications of linked list, Applications of doubly linked list. **Polynomial manipulation** - Polynomial manipulation, Addition of two polynomials, Multiplication of two polynomials. **Stack** - Introduction to stack, Representation of stack using array, Implementation of stack, Application of stack. **Evaluating arithmetic expressions - other applications** - Balancing Symbols, What is expression?, Conversion of infix to postfix expression, Evaluating postfix expression. **Queue ADT** - Introduction to Queue, Queue abstract data types, Example of operation on queue, Array implementation, Routine of array implementation. **Circular queue** - Circular queue, Implementation of circular queue using arrays, Creating and Initializing circular queue, Empty circular queue, Full circular Queue, Program for circular queue. **Double ended queues** - Double ended queue, Algorithms. **Applications of queue** - Applications of queue.

UNIT IV - NON-LINEAR DATA STRUCTURES

General Tree - Introduction, Definition of Tree, Tree Terminology. **Binary Tree** - Introduction to Binary Tree, Definition of Binary Tree, Representation of Binary Tree, Operations on Binary Tree, Applications of Binary Tree. **Binary tree representation and traversals** - Traversing a Binary Tree, Binary Tree Traversal, Algorithm - Preorder Traversal, Algorithm – In-order Traversal, Algorithm – Post order Traversal, Algorithm - Breadth First Tree Traversal. **Graphs** - Introduction to graphs, Directed and undirected graph. **Graph traversals** - Graph traversals, Depth First Search(DFS), Breadth First Search(BFS). **Representation of Graphs** - Representation of graphs, Adjacency matrix, Adjacency list. **Connected components** - Connected components.

UNIT V - SORTING AND SEARCHING

Introduction to sorting - Introduction to sorting, O notation, Efficiency of sorting. **Insertion sort** - Insertion sort, Program for insertion sort, Complexity analysis of insertion sort. **Quick sort** - Quick sort, Techniques of quick sort, Program for quick sort. **Merge sort** - Merge sort, Program for merge sort, Example of merge sort. **Searching: Linear search** - Introduction to searching, Linear search, Example -

Content Product

Detailed syllabus

linear search, Algorithm for linear search(unsorted array). General Requirements for Linear Search.

Binary search - Introduction to binary search, Binary search algorithm, Binary search - example, Performance of binary search, General requirements for binary search.