

# Content Product

## Detailed syllabus

### PROGRAMMING AND DATA STRUCTURES II

#### UNIT I - DATA ABSTRACTION AND OVERLOADING

**Why Object-** Why C++, Structure of C++ programs, Object oriented programming, Limitations of POP, 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). **Data Abstraction** - Data Abstraction. **Data Encapsulation** - Data Encapsulation. **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. **Constructors and destructors** - Class constructors, Constructors, Types of constructors, Copy constructors, Overloaded constructors, Dynamic constructors, Explicit constructors, Destructors. **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. **Static members – constant members** - Static data member, Static member functions, Constant member function. **Pointers** - Pointer, Creating pointers to variables, Creating pointers to an array, Creating pointers to functions, Creating pointers to objects, Pointers to objects, Program to illustrate pointers to objects, Program to illustrate array of pointers to objects, Formatted console I/O operations, Benefits of 'this' pointer, Example program of 'this' pointer. **Storage classes** - Storage classes, Auto storage class, Register storage class, Static storage class, Extern storage class, Mutable storage class. **Functions** - Introduction, Pre defined function, User defined functions, Function defining, Function declaration, Function call, Parameters. **Function as arguments** -Introduction, Function with no arguments and no return types, Function with arguments and no return types, Function with arguments and with return types, Function with no arguments and with return types, Parameter Passing Methods, Call by value, Call by reference, Call-by-Value Vs Call-by-Reference.

#### UNIT II - OBJECT ORIENTED PROGRAMMING CONCEPTS

**String Handling-** Introduction, Definition and Operations of String, Library Functions, Commonly used string constructors, Creating a string objects, Other ways to declare string objects, Example 1, Example 2, Example 3. **Copy constructors** - Copy constructors. **Polymorphism** - Polymorphism, Compile time and run time polymorphism, Introduction to virtual functions, 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. **Operator 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** - Function overloading. **Memory allocation** - Static memory allocation, Dynamic memory allocation, New operator, Delete operator, Program to illustrate deleting of objects using delete operators. **Nested classes** - Nested classes. **Inheritance** - Introduction to inheritance, Inheritance terminologies, Access control over the members, Access rights of derived

## Detailed syllabus

class, Relation between base class and derived class, Advantages of inheritance, Defining a class hierarchy, Visibility modes (access control), 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.

### UNIT III - C++ PROGRAMMING ADVANCED FEATURES

**Abstract class** - Abstract class. **Exception handling** - Introduction to exception handling, Basics of exception handling, Levels of exceptions, Exception handling mechanism, Catch handler, Execution of try-catch, Throwing exception in function, Throw and catch mechanism, Multiple catch statement, Specifying exception, Exceptions in constructors and destructors, Stack unwinding, Exceptions and inheritance, Rethrowing exceptions, Uncaught exception, Terminate function, Unexpected function. **Standard libraries** - Standard libraries, I/O Library, String library, Math library, Time library, Memory library. **Templates** - Templates, Difference between templates and macros, Need for templates, Types of templates, Guidelines for templates. **Class template** - Class template, Syntax for class template instantiation, Class template - syntax, Template arguments, Creating classes based on templates, Creating classes, Class template for stack data structure, Program to create template based stack which stores data of type integer and float. **Function templates** -C++ function template - approaches, Function template - syntax, Invalid declarations of function templates, Creating function based templates, Templates with non - type arguments, A more comprehensive example. **Generic programming and STL** - Introduction, Generic programming, OOP vs. Generic programming, Concept and model, STL components overview, Containers, Iterators, Algorithms, Function objects (functor), Functor allocators, Functor adapters. **Parameterizing the class** - Parameterizing the class, Example of parameterizing the class. **File handling concepts** - Introduction to file handling, File input and output stream, Need of file and file operations, File modes, String I/O to the files, Sequential input and output operations, Writing object to the file, Reading object from the file, Working with multiple files, Reading from two files simultaneously, Binary file, Reading and writing a class object, Random access file operation, Positioning the file pointer, Need for random access file, Object serialization.

### UNIT IV - ADVANCED NON-LINEAR DATA STRUCTURES

**Tree Structures** - Difference between linear and non – linear, Non – linear data structures, Introduction, Terminology, Attributes of the tree, Representation of trees, Expression tree, Parenthetical listing. **AVL trees** - Introduction, Representation of AVL trees, Insertion, Deletion, Pros and Cons of AVL trees. **B-Trees** - Introduction, Definition, Operations on B-Tree, Insertion, Deletion, Applications, advantage and disadvantage. **Red-Black trees** - Introduction to red-black tree, Rotations in red-black tree, Inserting in red-black tree, Algorithm for insertion, Deleting in red-black tree, Algorithm for deletion. **Splay tree** - Introduction to splay tree, Bottom up splay tree, Top-down splay trees, Splay tree operations, Advantage and disadvantages of splay tree, Applications of splay tree. Binomial. **Heaps – Fibonacci Heaps** - Binomial heaps, Fibonacci heaps, Disjoint sets, Path compression, Union by Rank. **Amortized Analysis** - Amortized Analysis, Accounting method, Potential method, Aggregate analysis.

# Content Product

## Detailed syllabus

### UNIT V – GRAPHS

**Representation of Graphs** - Introduction to graphs, Directed graphs, Representation of graphs, Adjacency matrix, Adjacency list. **Graphs traversals** -Graphs traversals, Depth First Search(DFS), Breadth First Search(BFS). **Topological sort** - Topological order traversal, Topological sort, Topological Sort: Complexity. **Spanning tree** - Spanning tree, Prim's algorithm, Kruskal's algorithm, Application of minimum spanning tree, Correctness of kruskal's algorithm, Correctness of prim's algorithm. **Shortest-path algorithms** - Definitions, Unweighted shortest path, Shortest path special cases, Shortest path Dijkstra's algorithm, Transitive closure, Floyd-Warshall's algorithm. **Bellman-Ford algorithm** - Bellman-Ford algorithm.