

C++ Programming

Ice Certification Program , CourseDuration :-(60Hrs)

UNIT -1 INTRODUCTION TO C++

- ❖ The history of c++
- ❖ Compilation steps
- ❖ Advantages and pretensions of c++
- ❖ Various programming technique
- ❖ Difference between structured programming language & object oriented programming Language.
- ❖ Definition of oop
- ❖ Features of oops are the following
- ❖ Data abstraction
- ❖ Encapsulation
- ❖ Polymorphism
- ❖ Inheritance
- ❖ Exception handling
- ❖ A simple c++ program
- ❖ Stream
- ❖ Standard input stream
- ❖ Standard output stream
- ❖ Generic classes
- ❖ Templates
- ❖ How c++ compilation works
- ❖ Commenting your programs
- ❖ Types of comments
- ❖ Using comments
- ❖ Variables
- ❖ Simple input/output constructs
- ❖ Variable types
- ❖ Names
- ❖ New operator
- ❖ Delete operator
- ❖ Operators
- ❖ Arithmetic operators
- ❖ Relational operators
- ❖ Logical operators
- ❖ Bitwise operators
- ❖ Increment/decrement operators
- ❖ Assignment operator
- ❖ Difference between c & c++
- ❖ Difference between delete & free
- ❖ Difference between new & malloc

UNIT 2: CLASSES AND OBJECTS

- ❖ Difference between structure & class
- ❖ Abstract data type
- ❖ Generic abstract data types
- ❖ Methods
- ❖ Classes
- ❖ Features of a class
- ❖ Objects
- ❖ Examples of objects
- ❖ Creation of objects:
- ❖ Accessing class members
- ❖ Program
- ❖ Abstraction
- ❖ How types of abstraction differs:
- ❖ Reasons for the need of abstraction
- ❖ Encapsulation
- ❖ Features of encapsulation:
- ❖ State
- ❖ Behavior
- ❖ Identity
- ❖ Constructors
- ❖ Types of constructor
- ❖ Destructors
- ❖ General syntax of destructors
- ❖ Instantiation of object
- ❖ Garbage collection
- ❖ What does libgc do?
- ❖ Common memory problems
- ❖ Memory leaks
- ❖ Premature frees
- ❖ Dynamic memory allocation
- ❖ What are memory management operators?
- ❖ New operator:
- ❖ Delete operator:
- ❖ Abstract classes

UNIT - 3 INHERITANCES

- ❖ **Objective**
- ❖ **Definition (inheritance) –**
- ❖ **Types of inheritance are as follows:-**
- ❖ **Single inheritance**
- ❖ **Multiple inheritance**
- ❖ **Hierarchical inheritance - one super class, many sub classes.**
- ❖ **Multilevel inheritance - derived from a derived class**
- ❖ **Hybrid inheritance - more than two types of inheritance.**

- ❖ Class hierarchy
- ❖ Derivation – public, private & protected
- ❖ Visibility mode
- ❖ Aggregation
- ❖ Composition
- ❖ Generalization

UNIT - 4 POLYMORPHISM

- ❖ Definition
- ❖ Features and advantages of the concept of polymorphism:
- ❖ Types of polymorphism:
- ❖ Virtual function -
- ❖ Example of virtual function rules
- ❖ For virtual functions pure
- ❖ Virtual functions parametric
- ❖ Polymorphism
- ❖ Overloadable operators
- ❖ Operators that cannot be overloaded

UNIT -5 GENERIC FUNCTIONS

- ❖ Objective
- ❖ Generic functions
- ❖ Class template with multiple parameters
- ❖ Example of template class with two generic data type

UNIT – 6 STREAMS AND FILES

- ❖ Objective
- ❖ C++ streams
- ❖ Standard input (stdin)
- ❖ Standard output (stdout)
- ❖ Standard error (stderr)
- ❖ C++'s predefined streams
- ❖ Unformatted i/o operations
- ❖ Formatted console i/o operations
- ❖ ios class functions and flags
- ❖ Defining display field width
- ❖ Setting precision
- ❖ Filling and padding
- ❖ ios formatting flags
- ❖ Manipulators
- ❖ Custom/user-defined manipulators
- ❖ Space.cpp
- ❖ What is a file?
- ❖ Hierarchy of file stream classes
- ❖ Opening and closing a file
- ❖ Opening and closing of files explicitly
- ❖ Files modes

- ❖ **Ascii and binary files**
- ❖ **Binary file**
- ❖ **Write () and read () functions**

UNIT -7 NAMESPACE, EXCEPTION HANDLING & STL

- ❖ Container adapters
- ❖ Algorithms
- ❖ Iterators
- ❖ Associative containers
- ❖ Sets and multisets
- ❖ Maps and multimaps
- ❖ Namespace fundamentals
- ❖ Using
- ❖ Unnamed namespaces
- ❖ Exception handling fundamentals
- ❖ Exception handling constructs
- ❖ Using multiple catch statements
- ❖ Rethrowing an exception
- ❖ Introduction to the stl (standard template language)
- ❖ Containers
- ❖ Sequence containers
- ❖ Associative containers
- ❖ Member functions
- ❖ Container adapters
- ❖ Algorithms
- ❖ Iterators
- ❖ Sequence containers
- ❖ Vectors
- ❖ Lists
- ❖ Deques
- ❖ Associative containers
- ❖ Sets and multisets
- ❖ Maps and multimaps, Author Name
- ❖ Media Type, Author Name