Teaching Guidelines for

C++ Programming

PG-DAC August 2025

Duration: 72 hours (36 theory hours + 36 lab hours)

Objective: To learn object oriented programming using C++

Prerequisites: Knowledge of computer fundamentals

Evaluation: 100 marks
Weightage: CCEE – 40%, Lab exam – 40%, Internals – 20%

Text Book:

- Object Oriented Programming in C++ by Rajesh K. Shukla / Wiley

References:

- Object Oriented Programming with C++ by E Balagurusamy / McGraw Hill
- The C++ Programming Language, Bjarne Stroustrup

(Note: Each Session is of 2 hours theory and 2 hours lab unless mentioned otherwise)

Session 1: Getting Started
Lecture:

- Installation and Setup development environment
- The need of C++
- Features of C++
- C++ versus C
- History of C++
- Writing your first C++ program

Lab:
Write different C++ programs to

- Print Hello World
- Add two numbers/binary numbers/characters
- Calculate compound interest
- Calculate power of a number
- Swap two numbers
- Calculate area of rectangle

Sessions 2 & 3: Beginning with C++
Lecture:

- C++Program structure
- Introduction of advanced C++ concepts and feature of C++ 17
- C++ Tokens
- Initialization
- Static Members
- Constant Members
- Expressions

Operators

- Arithmetic Operator
- Relational Operator
- Logical Operator
- Unary Operator
- Ternary Operator
- Assignment Operator

Lab:

- Write a Student class and use it in your program. Store the data of 10 students and display the sorted data according to their roll numbers, dates of birth, and total marks.
- Implement all C++ operators
- Declare members and implement in your programs.

Session 4: Conditional and Looping Statements

Lecture:

- If, else if, switch
- for loop
- while loop
- do while loop
- Jump statement (break, continue& returnkeyword)
- Arrays
- Declaration and initialization of an array
- 1-D and 2-D arrays

Lab:

- Implement all control structures through your program
- Implement a program which accepts command line arguments from main function.

Session 5: Functions in C++

Lecture

- Different forms of functions
- Function prototyping
- Call by Reference
- Inline Functions
- Math library functions etc.

Lab:

- Implement functions through your program
- Declare function and call it by reference and note the observations
- Implement Inline functions in your program

Sessions 6 & 7: Memory Management and Pointers

Lecture

- Introduction to memory management in C++
- Pointers in C++
- Arrays using pointers
- Enumeration
- Typedef
- Using New operator
- Class pointer
- this pointer
- Comparison of new over malloc, calloc and realloc, etc.
- Memory freeing using Delete operator

Lab:

- Assignments using pointers, arrays of pointers
- Assignments on passing pointers in functions
- Using pointers, write your own functions for the following:
    - String comparison
    - String concatenate
    - String copy
    - String length

Note: Do not include <string.h> in your program and implement Delete operator in your program.

Session 8: OOP Concepts

Lecture

- Discussion on object oriented concepts
- Classes and Objects, Access Specifiers, Overloading, Inheritance,Polymorphism
- Namespaces

Lab:

- Write a student class and use it in your program. Store the data of 10 students and display the sorted data according to their roll numbers, dates of birth, and total marks.

Session 9: Constructors and Destructor

Lecture

- Constructors
- Parameterized constructors
- Multiple constructors in class
- Dynamic initialization of objects
- Copy Constructors
- Destructor

Lab:

- Implement constructor and destructors through your program
- Write a program to implement inner class in C++

Session 10: Inheritance – extending class

Lecture

- Types of inheritance
- Single inheritance
- Multiple inheritance
- Multilevel inheritance
- Hierarchical inheritance
- Hybrid inheritance, etc.
- Virtual base class
- Constructors in derived class

Lab:

- Design a hierarchy of computer printers. Use multiple inheritance in your hierarchy. Also use friend functions and classes in your program.

Session 11: Polymorphism

Lecture

- Types of Polymorphism
- Overloading functions
- Overloading Operators
- Friend functions

- Constant functions

Lab:

- Write Date and Time classes that allow you to add, subtract, read and print simple dates in dd/mm/yyyy and time in hh:mm:ss formats. Use function overloading in your program.
- Assignments to overload =, ==, +, ++, --, <<, >> and [ ]operators.

Session 12: Virtual Functions and Abstract Class

Lecture

- Run Time Polymorphism
- Virtual Functions and Pure virtual functions
- dynamic_cast, static_cast,const_cast, reinterpret_cast
- Interfaces
- Abstract class

Lab:

- Implement Abstract classes in your program
- Using virtual and pure virtual functions implement hierarchy of computer printers
- Implement diamond problem with real life example

Session 13: Exception Handling

Lecture

- Exception Handling Introduction
- Exception handling – throwing, catching, re-throwing anexception
- Specifying exceptions etc.

Lab:

- Implement exceptions and do proper management through your program
- Implement Custom exception class

Session 14: Managing Console I/O Operations

Lecture

- Introduction
- C++ streams
- C++ stream classes
- Unformatted I/O operations
- Formatted I/O operations
- Managing output with manipulators

Lab:

- Implement console I/O operations through your program.

Session 15: File Handling in C++

Lecture

- Definition of file
- File handling in C++
- Doing read, write operation in files

Lab:

- Assignments on files doing different operations

Session 16: Templates

Lecture

- Introduction to Templates
- Function Templates
- Class Templates

Lab:

- Assignments on templates

Sessions 17 & 18: STL and RTTI

Lecture

- Introduction to C++ Standard Library
- Working with Stack, Vector, Queue, Map
- Introduction to RTTI (Run-Time Type Information) in C++

Lab:

- Assignments on STL Library