Suggested Teaching Guidelines for

OOPs with C++ Programming – *PG-DAC August 2018*

Duration: 30 class room hours + 30 lab hours (**60hrs**)

Objective: To reinforce knowledge of Object Oriented Concepts and C++ Programming.

Prerequisites: Knowledge of C Programming

Evaluation method: Theory exam– 40% weightage
Lab exam – 40% weightage
Internal exam– 20% weightage

List of Books / Other training material

Test Book:

1. C++ Primer Plus by Stephen Prata / Pearson

Reference:

1. Thinking in C++ by Bruce Eckel
2. The C programming Language by Kernighan and Retchie
3. Advanced Unix Programming by Rochkind
4. The C++ Programming Language, Bjarne Stroustrup;
5. Object-oriented Analysis And Design Using Umlan Introduction To Unified Process And Design Patterns 1st Edition by Mahesh P. Matha / PHI

Session 1 & 2: (Linux Command and Revision of C Programming)

Lecture

- Linux introduction
- Basic Linux commands required for writing and executing C++ programs
- Revision of C Programming
    - Pointers
    - Functions (Call by value and reference)
    - Recursion
    - Arrays using Pointers
    - Structures
    - Union
    - Enumeration and Typedef
    - File handling

Assignment – Lab:

- Practice of Linux commands
- Writing different C programs based on functions, structures, file and executing them

Session 3: (OOP Concepts)

Lecture

- Difference between C and C++
- Discussion on Object oriented concepts
- Classes and Objects, Access Specifiers, Overloading, Inheritance, Polymorphism
- Constructors and Destructors

- Namespaces

Assignment – Lab:

- Write a Student class and use it in your program. Store the data of ten students and display the sorted data according to their roll numbers, date of births, and total marks.

Session 4: Beginning with C++

Lecture

- C++ Tokens
- Initialization
- C++ Operators
- Static Members
- Constant Members
- Expressions
- Control Structure

Assignment – Lab:

- Implement all C++ operators
- Declare members and implement in your programs.
- Implement all control structures through your program

Session 5: Functions in C++

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

Assignment – Lab:

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

Session 6 & 7: Memory Management and pointers

Lecture

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

Assignment – 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: Constructor and Destructor
Lecture

- Constructors
- Parameterized constructors
- Multiple constructors in class
- Dynamic initialization of objects
- Copy Constructors
- Encapsulating into an object
- Destructors
- Associations
- Inner Classes

Assignment – Lab:

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

Session 9 & 10: Inheritance, Virtual Functions, Polymorphism
Lecture

- Inheritance, Virtual Functions, Polymorphism
- Inheritance
- Type of Inheritances
- Run Time Polymorphism
- Polymorphism
- Overloading functions
- Overloading Operators
- Friend Functions
- Virtual Functions and Pure virtual functions

Assignment – Lab:

- Write Date and Time classes that allows you to add, subtract, read and print simple dates in dd/mm/yyyy and time in hh:mm:ss forms. Use function overloading in your program.
- Assignments to overload =, ==, +, ++, --, <<, >> and [ ] operators.
- Design a hierarchy of computer printers. Use multiple inheritance in your hierarchy. Also use friend functions and classes in your program.
- Using virtual and pure virtual functions implement hierarchy of computer printers.

Session 11: Inheritance – extending class
Lecture

- Introduction

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

Assignment – Lab:

- Implement Abstract classes in your program

Session 12: Exception Handling

Lecture

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

Assignment – Lab:

- Implement exceptions and do proper management through your program

Session 13: Managing Console I/O Operations

Lecture

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

Assignment – Lab:

- Implement console I/O operations through your program

Session 14: File Handling in C++

Lecture

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

Assignment – Lab:

- Assignments on files doing different operations.

Session 15: Introduction to Advanced C++ Concepts

Lecture

- Object Design and Templates
- STL (Standard Type Libraries)

- RTTI (Run Time Type Identification)
- Advanced Typecasting
- new data types
- new operators
- class implementation
- namespace scope
- operator keywords
- new headers
- C++ Containers

Assignment – Lab:

- Find the number of students who are passes or failed using MAP.
- Find the prime numbers from 2 to n using sieves algorithm, use SET
- Implement the Run time polymorphism using RTTI