OBJECT ORIENTED PROGRAMMING

ENCT 151

Lecture : 3 **Year : I**
Tutorial : 1 **Part : II**
Practical : 3

Course Objectives:
To provide students with foundation in understanding and applying principles of object oriented programming. Emphasis will be given on developing object oriented programming skills using C ++.

1 Introduction to Object Oriented Programming (3 hours)

1.1 History of programming languages
1.2 Need of object oriented programming(OOP)
1.3 Object oriented programming versus procedure oriented programming
1.4 Concepts of object oriented programming
1.5 Popular object oriented languages
1.6 Advantages of OOP
1.7 Disadvantages of OOP

2 Basics of C++ Programming (5 hours)

2.1 C++ Program structure
2.2 Character set and tokens
2.3 Variable declaration and expression
2.4 Data types
2.5 Type conversion and casting
2.6 User defined constant const
2.7 Reference variables
2.8 Conditions and looping
2.9 Namespace scope
2.10 Functions
  2.10.1 Function overloading
  2.10.2 Inline function
  2.10.3 Default arguments
  2.10.4 Pass by reference and return by reference
2.11 Array, pointer and string
2.12 Structure and unions
2.13 Enumeration
2.14 Dynamic memory allocation

3 Objects and Classes (7 hours)

3.1 C++ classes
3.2 Objects and the member access
3.3 Relation of object, class and memory
3.4 Defining member function
3.5 Defining outer function inline
3.6 Objects as member
3.7 Constructors and destructors
3.8 Object as function arguments
3.9 Returning objects from functions
3.10 Array of objects
3.11 Pointer to objects
3.12 Dynamic memory allocation for objects
3.13 Dynamic constructors
3.14 Pointer
3.15 Static data member and static member function
3.16 Constant member functions and constant objects
3.17 Friend functions and friend classes

4 Operator Overloading (5 hours)

4.1 Overloadable and non- overloadable operators
4.2 Syntax of operator overloading
4.3 Operator overloading using member operator functions
4.4 Operator overloading using non member functions
4.5 Unary operator overloading
4.6 Binary operator overloading
4.7 Type conversion between objects
4.8 Explicit constructors

5 Inheritance (5 hours)

5.1 Base class and derived class
5.2 protected access specifier
5.3 Derived class declaration
5.4 Is_a relation and Has_a relation
5.5 Public, protected and private inheritance
5.6 Member overriding
5.7 Forms of inheritance
5.8 Constructors in derived class
5.9 Destructor in derived class
5.10 Need of virtual base class

6 Virtual Functions (4 hours)

6.1 What is virtual function
6.2 Need of virtual function
6.3 Pointer to derived class
6.4 Array of pointers to base class
6.5 Pure virtual functions and abstract class
6.6 Virtual destructors
6.7 Reinterpret cast operator
6.8 Run-time type information

7 Stream Computation (6 hours)

7.1 Input/output stream class hierarchy
7.2 Testing stream errors
7.3 Unformatted input/output and formatted input/output
7.4 Stream operator overloading
7.5 File input/output with streams
7.6 File stream class hierarchy
7.7 ASCII and binary files
7.8 Read/Write from file
7.9 Sequential access to file
7.10 Random access to file
7.11 File access pointers and their manipulators
7.12 Testing errors during file operations
7.13 File input/output with member functions

8 Templates (6 hours)

8.1 Function template
8.2 Overloading function template
8.3 Class template
8.4 Derived class template
8.5 Introduction to standard template library
    8.5.1 Components of STL
    8.5.2 Container
    8.5.3 Iterators
    8.5.4 Algorithms

9 Exception Handling (4 hours)

9.1 Basics of exception handling
9.2 Advantage over conventional error handling
9.3 Exception handling mechanism
9.4 Multiple handlers
9.5 Catching all exceptions

9.6 Rethrowing exception
9.7 Exception with arguments
9.8 Exceptions specification for function
9.9 Exceptions in constructors and destructors
9.10 Handling uncaught exceptions
9.11 Handling unexpected exception

Tutorial **(15 hours)**

After completing each chapter some problems are solved and students are asked to solve programming problems with the teacher's assistance.

Assignment

Appropriate assignment problems are given to students after the completion of each chapter.

Practical **(45 hours)**

There will be nine laboratory exercises encompassing the entire course content. At the end of the course, students are required to submit a programming project demonstrating the application of object-oriented programming concepts.

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as possible as indicated in the table below:

Chapter	Hours	Marks Distribution*
1	3	4
2	5	7
3	7	9
4	5	7
5	5	7
6	4	5
7	6	8
8	6	8
9	4	5
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Schildt, H. (2015). C++: The Complete Reference. McGraw-Hill Education.
2. Baral, D.S., Baral, D. (2010). Secrets of Object Oriented Programming in C++. Bhundipuran Prakasan.
3. Lafore, R. (2002). Object Oriented Programming in C++. Sams Publishing.
4. Deitel, H.M., Deitel, P.J. (2007). C++ How to Program. Prentice Hall.