OBJECT ORIENTED PROGRAMMING | SYLLABUS | MARKING SCHEME | IOE

OBJECT ORIENTED PROGRAMMING

CT 501

Lecture : 3
Year : II
Tutorial : 0
Part : I
Practical : 3

Course Objective:

The objective of the course is to familiarize students with the C++ programming language and use the language to develop pure object oriented programs.

1. Introduction to Object Oriented Programming (3 hours)

1.1 Issues with Procedure Oriented Programming
1.2 Basic of Object Oriented Programming (OOP)
1.3 Procedure Oriented versus Object Oriented Programming
1.4 Concept of Object Oriented Programming
1.4.1 Object
1.4.2 Class
1.4.3 Abstraction
1.4.4 Encapsulation
1.4.5 Inheritance
1.4.6 Polymorphism
1.5 Example of Some Object Oriented Languages
1.6 Advantages and Disadvantages of OOP

2. Introduction to C++ (2 hours)

2.1 The Need of C++
2.2 Features of C++

2.3 C++ Versus C
2.4 History of C++

3. C++ Language Constructs (6 hours)

3.1 C++ Program Structure
3.2 Character Set and Tokens
3.2.1 Keywords
3.2.2 Identifiers
3.2.3 Literals
3.2.4 Operators and Punctuators
3.3 Variable Declaration and Expression
3.4 Statements
3.5 Data Type
3.6 Type Conversion and Promotion Rules
3.7 Preprocessor Directives
3.8 Namespace
3.9 User Defined Constant const
3.10 Input/Output Streams and Manipulators
3.11 Dynamic Memory Allocation with new and delete
3.12 Condition and Looping
3.13 Functions
3.13.1 Function Syntax
3.13.2 Function Overloading
3.13.3 Inline Functions
3.13.4 Default Argument
3.13.5 Pass by Reference
3.13.6 Return by Reference
3.14 Array, Pointer and String
3.15 Structure, Union and Enumeration

4. Objects and Classes (6 hours)

4.1 C++ Classes

4.2 Access Specifiers
4.3 Objects and the Member Access
4.4 Defining Member Function
4.5 Constructor
4.5.1 Default Constructor
4.5.2 Parameterized Constructor
4.5.3 Copy Constructor
4.6 Destructors
4.7 Object as Function Arguments and Return Type
4.8 Array of Objects
4.9 Pointer to Objects and Member Access
4.10 Dynamic Memory Allocation for Objects and Object Array
4.11 this Pointer
4.12 static Data Member and static Function
4.13 Constant Member Functions and Constant Objects
4.14 Friend Function and Friend Classes

5. Operator Overloading (5 hours)

5.1 Overloadable Operators
5.2 Syntax of Operator Overloading
5.3 Rules of Operator Overloading
5.4 Unary Operator Overloading
5.5 Binary Operator Overloading
5.6 Operator Overloading with Member and Non Member Functions
5.7 Data Conversion: Basic – User Defined and User Defined – User Defined
5.8 Explicit Constructors

6. Inheritance (5 hours)

6.1 Base and Derived Class
6.2 protected Access Specifier
6.3 Derived Class Declaration
6.4 Member Function Overriding

6.5 Forms of Inheritance: single, multiple, multilevel, hierarchical, hybrid, multipath
6.6 Multipath Inheritance and Virtual Base Class
6.7 Constructor Invocation in Single and Multiple Inheritances
6.8 Destructor in Single and Multiple Inheritances

7. Polymorphism and Dynamic Binding (4 hours)

7.1 Need of Virtual Function
7.2 Pointer to Derived Class
7.3 Definition of Virtual Functions
7.4 Array of Pointers to Base Class
7.5 Pure Virtual functions and Abstract Class
7.6 Virtual Destructor
7.7 reinterpret_cast Operator
7.8 Run-Time Type Information
7.8.1 dynamic_cast Operator
7.8.2 typeid Operator

8. Stream Computation for Console and File Input /Output (5 hours)

8.1 Stream Class Hierarchy for Console Input /Output
8.2 Testing Stream Errors
8.3 Unformatted Input /Output
8.4 Formatted Input /Output with ios Member functions and Flags
8.5 Formatting with Manipulators
8.6 Stream Operator Overloading
8.7 File Input/output with Streams
8.8 File Stream Class Hierarchy
8.9 Opening and Closing files
8.10 Read/Write from File
8.11 File Access Pointers and their Manipulators
8.12 Sequential and Random Access to File
8.13 Testing Errors during File Operations

9. Templates (5 hours)

9.1 Function Template
9.2 Overloading Function Template
9.2.1 Overloading with Functions
9.2.2 Overloading with other Template
9.3 Class Template
9.3.1 Function Definition of Class Template
9.3.2 Non-Template Type Arguments
9.3.3 Default Arguments with Class Template
9.4 Derived Class Template
9.5 Introduction to Standard Template Library
9.5.1 Containers
9.5.2 Algorithms
9.5.3 Iterators

10. Exception Handling (4 hours)

10.1 Error Handling
10.2 Exception Handling Constructs (try, catch, throw)
10.3 Advantage over Conventional Error Handling
10.4 Multiple Exception Handling
10.5 Rethrowing Exception
10.6 Catching All Exceptions
10.7 Exception with Arguments
10.8 Exceptions Specification for Function
10.9 Handling Uncaught and Unexpected Exceptions

Practical:

There will be about 12 lab exercises covering the course. At the end of the course students must complete a programming project on object oriented programming with C++.

References :

1 Robert Lafore, “Object Oriented Programming in C++”, 4th Edition 2002, Sams Publication
2 Daya Sagar Baral and Diwakar Baral, “The Secrets of Object Oriented Programming in C++”, 1st Edition 2010, Bhundipuran Prakasan
3 Harvey M. Deitel and Paul J. Deitel, “C++ How to Program”, 3rd Edition 2001, Pearson Education Inc.
4 D. S. Malik, “C++ Programming”, 3rd Edition 2007, Thomson Course Technology
5 Herbert Schildt, “C++: The Complete Reference”, 4th Edition 2003, Tata McGraw Hill

Evaluation Scheme:

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

Chapters Hours Marks distribution*

Chapters	Hours	Marks distribution*
1,2,4	11	20
3	6	10
5	5	10
6	5	10
8	5	10
7,9,10	13	20
Total	**45**	**80**

*There may be minor deviation in marks distribution