Object Oriented Programming

Course Objective:
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)
    a. Issues with Procedure Oriented Programming
    b. Basic of Object Oriented Programming (OOP)
    c. Procedure Oriented versus Object Oriented Programming
    d. Concept of Object Oriented Programming
        i. Object
        ii. Class
        iii. Abstraction
        iv. Encapsulation
        v. Inheritance
        vi. Polymorphism
    e. Example of Some Object Oriented Languages
    f. Advantages and Disadvantages of OOP

-

2. Introduction to C++(2 hours)
    a. The Need of C++
    b. Features of C++
    c. C++ Versus C
    d. History of C++

-

3. C++ Language Constructs(6 hours)
    a. C++ Program Structure
    b. Character Set and Tokens
        i. Keywords
        ii. Identifiers
        iii. Literals
        iv. Operators and Punctuators
    c. Variable Declaration and Expression
    d. Statements
    e. Data Type
    f. Type Conversion and Promotion Rules
    g. Preprocessor Directives
    h. Namespace
    i. User Defined Constant
    j. Input/Output Streams and Manipulators
    k. Dynamic Memory Allocation with new and delete
    l. Condition and Looping
    m. Functions
        i. Function Syntax
        ii. Function Overloading
        iii. Inline Functions
        iv. Default Argument
        v. Pass by Reference
        vi. Return by Reference
    n. Array, Pointer and String
    o. Structure, Union and Enumeration

4. Objects and Classes(6 hours)
    a. C++ Classes
    b. Access Specifiers
    c. Objects and the Member Access
    d. Defining Member Function
    e. Constructor
        i. Default Constructor
        ii. Parameterized Constructor
        iii. Copy Constructor
    f. Destructors
    g. Object as Function Arguments and Return Type
    h. Array of Objects
    i. Pointer to Objects and Member Access
    j. Dynamic Memory Allocation for Objects and Object Array
    k. Pointer
    l. Static Data Member and Static Function
    m. Constant Member Functions and Constant Objects
    n. Friend Function and Friend Classes

5. Operator Overloading(5 hours)
    a. Overloadable Operators
    b. Syntax of Operator Overloading
    c. Rules of Operator Overloading
    d. Unary Operator Overloading
    e. Binary Operator Overloading
    f. Operator Overloading with Member and Non Member Functions
    g. Data Conversion: Basic–User Defined and User Defined–User Defined
    h. Explicit Constructors

6. Inheritance(5 hours)
    a. Base and Derived Class
    b. protected Access Specifier
    c. Derived Class Declaration
    d. Member Function Overriding
    e. Forms of Inheritance: Single, Multiple, Multilevel, Hierarchical, Hybrid, Multipath
    f. Multipath Inheritance and Virtual Base Class
    g. Constructor Invocation in Single and Multiple Inheritances
    h. Destructor in Single and Multiple Inheritances

7. Polymorphism and Dynamic Binding(4 hours)
    a. Need of Virtual Function
    b. Pointer to Derived Class
    c. Definition of Virtual Functions
    d. Array of Pointers to Base Class
    e. Pure Virtual functions and Abstract Class
    f. Virtual Destructor
    g. Reinterpret_cast Operator
    h. Run-Time Type Information
        i. Dynamic_cast Operator
        ii. Type_id Operator

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

   a. Stream Class Hierarchy for Console Input/Output
   b. Testing Stream Errors
   c. Unformatted Input/Output
   d. Formatted Input/Output with IOS Member functions and Flags
   e. Formatting with Manipulators
   f. Stream Operator Overloading
   g. File Input/output with Streams
   h. File Stream Class Hierarchy
   i. Opening and Closing files
   j. Read/Write from File
   k. File Access Pointers and their Manipulators
   l. Sequential and Random Access to File
   m. Testing Errors during File Operations

9. Templates(5 hours)

   a. Function Template
   b. Overloading Function Template
      i. Overloading with Functions
      ii. Overloading with other Template
   c. Class Template
      i. Function Definition of Class Template
      ii. Non-Template Type Arguments
      iii. Default Arguments with Class Template
   d. Derived Class Template
   e. Introduction to Standard Template Library
      i. Containers
      ii. Algorithms
      iii. Iterators

10. Exception Handling(4 hours)

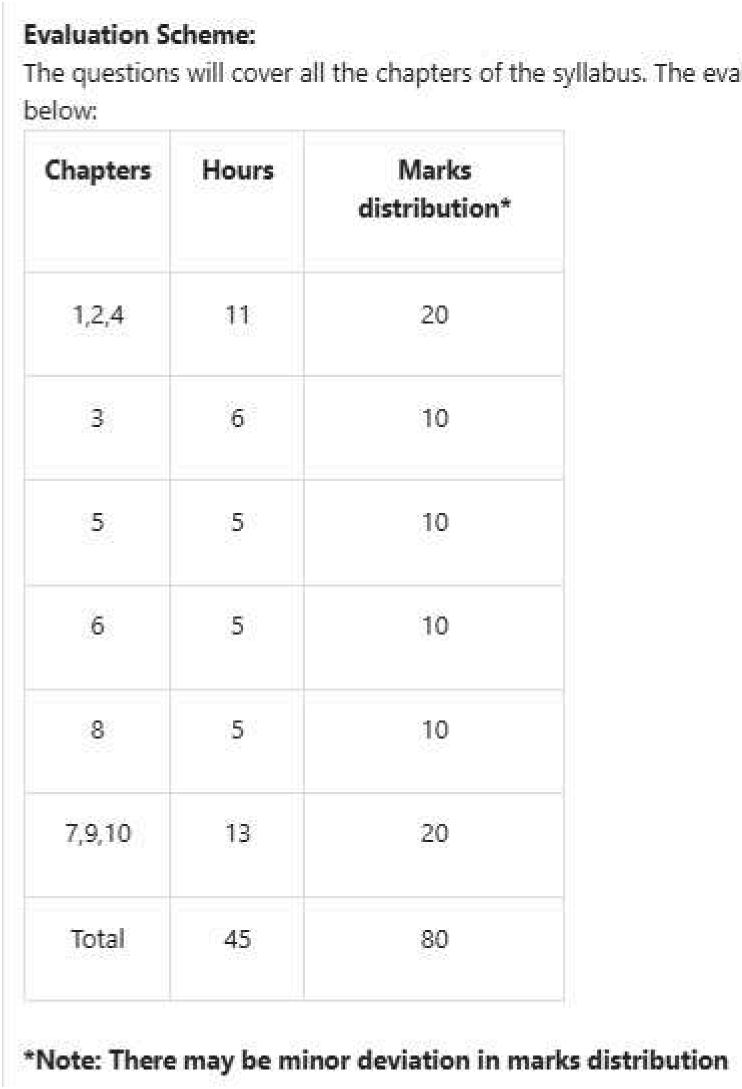
   a. Error Handling
   b. Exception Handling Constructs (try, catch, throw)
   c. Advantage over Conventional Error Handling
   d. Multiple Exception Handling
   e. Rethrowing Exception
   f. Catching All Exceptions
   g. Exception with Arguments
   h. Exceptions Specification for Function
   i. Handling 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*
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

***Note: There may be minor deviation in marks distribution**