**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)**
   1. Issues with Procedure Oriented Programming
   2. Basic of Object Oriented Programming (OOP)
   3. Procedure Oriented versus Object Oriented Programming
   4. Concept of Object Oriented Programming
      1. Object
      2. Class
      3. Abstraction
      4. Encapsulation
      5. Inheritance
      6. Polymorphism
   5. Example of Some Object Oriented Languages
   6. Advantages and Disadvantages of OOP
2. **Introduction to C++(2 hours)**
   1. The Need of C++
   2. Features of C++
   3. C++ Versus C
   4. History of C++
3. **C++ Language Constructs(6 hours)**
   1. C++ Program Structure
   2. Character Set and Tokens
      1. Keywords
      2. Identifiers
      3. Literals
      4. Operators and Punctuators
   3. Variable Declaration and Expression
   4. Statements
   5. Data Type
   6. Type Conversion and Promotion Rules
   7. Preprocessor Directives
   8. Namespace
   9. User Defined Constant
   10. Input/Output Streams and Manipulators
   11. Dynamic Memory Allocation with new and delete
   12. Condition and Looping
   13. Functions
       1. Function Syntax
       2. Function Overloading
       3. Inline Functions
       4. Default Argument
       5. Pass by Reference
       6. Return by Reference
   14. Array, Pointer and String
   15. Structure, Union and Enumeration
4. **Objects and Classes(6 hours)**
   1. C++ Classes
   2. Access Specifiers
   3. Objects and the Member Access
   4. Defining Member Function
   5. Constructor
      1. Default Constructor
      2. Parameterized Constructor
      3. Copy Constructor
   6. Destructors
   7. Object as Function Arguments and Return Type
   8. Array of Objects
   9. Pointer to Objects and Member Access
   10. Dynamic Memory Allocation for Objects and Object Array
   11. Pointer
   12. Static Data Member and Static Function
   13. Constant Member Functions and Constant Objects
   14. Friend Function and Friend Classes
5. **Operator Overloading(5 hours)**
   1. Overloadable Operators
   2. Syntax of Operator Overloading
   3. Rules of Operator Overloading
   4. Unary Operator Overloading
   5. Binary Operator Overloading
   6. Operator Overloading with Member and Non Member Functions
   7. Data Conversion: Basic–User Defined and User Defined–User Defined
   8. Explicit Constructors
6. **Inheritance(5 hours)**
   1. Base and Derived Class
   2. protected Access Specifier
   3. Derived Class Declaration
   4. Member Function Overriding
   5. Forms of Inheritance: Single, Multiple, Multilevel, Hierarchical, Hybrid, Multipath
   6. Multipath Inheritance and Virtual Base Class
   7. Constructor Invocation in Single and Multiple Inheritances
   8. Destructor in Single and Multiple Inheritances
7. **Polymorphism and Dynamic Binding(4 hours)**
   1. Need of Virtual Function
   2. Pointer to Derived Class
   3. Definition of Virtual Functions
   4. Array of Pointers to Base Class
   5. Pure Virtual functions and Abstract Class
   6. Virtual Destructor
   7. Reinterpret\_cast Operator
   8. Run-Time Type Information
      1. Dynamic\_cast Operator
      2. Type\_id Operator
8. **Stream Computation for Console and File Input/Output(5 hours)**
   1. Stream Class Hierarchy for Console Input/Output
   2. Testing Stream Errors
   3. Unformatted Input/Output
   4. Formatted Input/Output with IOS Member functions and Flags
   5. Formatting with Manipulators
   6. Stream Operator Overloading
   7. File Input/output with Streams
   8. File Stream Class Hierarchy
   9. Opening and Closing files
   10. Read/Write from File
   11. File Access Pointers and their Manipulators
   12. Sequential and Random Access to File
   13. Testing Errors during File Operations
9. **Templates(5 hours)**
   1. Function Template
   2. Overloading Function Template
      1. Overloading with Functions
      2. Overloading with other Template
   3. Class Template
      1. Function Definition of Class Template
      2. Non-Template Type Arguments
      3. Default Arguments with Class Template
   4. Derived Class Template
   5. Introduction to Standard Template Library
      1. Containers
      2. Algorithms
      3. Iterators
10. **Exception Handling(4 hours)**
    1. Error Handling
    2. Exception Handling Constructs (try, catch, throw)
    3. Advantage over Conventional Error Handling
    4. Multiple Exception Handling
    5. Rethrowing Exception
    6. Catching All Exceptions
    7. Exception with Arguments
    8. Exceptions Specification for Function
    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\*** |
| 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**