# Object Oriented Paradigm

**Credit Hours:** 3+1

**Course Objectives:** This course is a comprehensive hands-on introduction to object oriented programming in C++. Recognize features of object-oriented design and implement those using features of C++ that support effective modeling of the problem domain and reuse of code.

**Course Instructor**: Amina Mukhtar

**Prerequisites:** Programming Fundamentals

**OOP as Prerequisite**: Data Structures

**Text Book:** C++ How to Program by Paul Deitel and Harvey Deitel

**Reference Books:**

* [C++ Programming Language by Bjarne Stroustrup, 4th Edition](http://www.amazon.com/Programming-Absolute-Beginner-Mark-Lee/dp/1598638750/ref=sr_1_8?ie=UTF8&s=books&qid=1271830370&sr=1-8)
* Problem Solving with C++ by Walter Savtich, 9th Edition

**Syllabus:**

|  |  |
| --- | --- |
| **Lecture** | **Topics** |
| 1 | **Recap**: Pointers, Pointers in C++, Pointers Vs. Array (Pointer to an array, Passing Arrays to function, return Array from function) Array of Pointers, Introduction of STL |
| 2 | **Recap**: Passing Pointers to functions, Return Pointers from Function, References, References Vs. Pointers, References as parameters, Reference as return value, STL implementations (std::vector, std::string) |
| 3 | Course Introduction, Object Oriented Model, benefits and implications, Introduction to Objects, Example related to object |
| 4 | Classes, Concepts of Information Hiding, Encapsulation, Abstraction |
| 5 | Classes in C++, Syntax and Semantics, Member variables, Member functions, public and priavte access specifiers, Inline functions, Method Overlaoding |
| 6 | Constructors and member initializer list, Default Constructor, Constructor Overloading, Delegating constructors |
| 7 | Copy Constructor, Shallow copy, Deep copy |
| 8 | Destructors, Introduction to getter and setter methods, this pointer, Constant data members, constant member functions |
| 9 | Object creation and deletion, Object Life Time and Scope, Static Variables, Accessing Static data members and static Functions, Global Variables vs. Static Data Members, |
| 10 | Assignment vs. Initialization, Friend Functions, Operator overloading, Assignment Operator Overloading |
| 11 | Operator Overloading unary operator (++,--), Pre and Post, Binary Operator Overloading, Stream Insertion, Stream Extraction  **(Additional**: Operator Overloading(\*,->)) |
| 12 | Inheritance, Generalization, Sub Typing (Extension), Specialization (Restriction) |
| 13 | Inheritance in C++, “IS A” relationship, Inheritance UML Diagram, Public Inheritance Implementation in C++, Accessing Base class Member Functions in Derived class, Protected Access Specifier |
| 14 | Polymorphism, Polymorphism in OO Model, Advantages of Polymorphism, case study |
| 15 | Static vs Dynamic Binding, Virtual Functions, Overriding, Overloading vs Overriding, Virtual Specifier, override Specifier, final Specifier |
|  | **MID TERM** |
| 16 | Virtual Destructors, VTables, Dynamic Dispach |
| 17 | Inheritance (Public) - Case Study |
| 18 | Types of Inheritance, Purpose of Private and Protected Inheritance, Private Inheritance, Protected Inheritance in C++. |
| 19 | Multiple Inheritance, Implementation of Multiple Inheritance in C++, Issues in Multiple inheritance |
| 20 | Problems in Multiple Inheritance (Diamond Problem), virtual Inheritance |
| 21 | Inheritance (Public, Private, Protected) - Case Study |
| 22 | Association, Aggregation, Implementation of examples in C++ |
| 23 | Composition, Implementation of examples in C++, Difference between composition and aggregation |
| 24 | Abstract Classes, Concrete classes, Pure Virtual Functions |
| 25 | Generic Programming, Templates, Function Templates |
| 26 | Examples related to Function Templates, Multiple Type Arguments, |
| 27 | Class Templates, Examples related to Class Templates |
| 28 | Member Templates, Templates Specialization, Overloading Vs Templates |
| 29 | Error Handling, Exception Handling (throwing and catching Exception), Difference in Error and Exception, |
| 30 | C++ Standard Exceptions, Use of Exception Handling, Examples related to Exception Handling |

**Evaluation Criteria:**

|  |  |
| --- | --- |
| Quizzes (4) | 10 |
| Assignments (3) | 10 |
| Midterm | 25 |
| Final | 30 |
| Labs | 15 |
| Final Project | 10 |