

# 10B11CI311: Object Oriented Programming

**Course Credit: 4**

**Semester: III**

## Introduction

This course presents a conceptual and practical introduction to imperative and object oriented programming, exemplified by C++ and Java. As well as providing a grounding in the use of C++ and Java, the course will cover general principles of programming in imperative and object oriented frameworks. The course should enable you to develop programs that support experimentation, simulation and exploration in other parts of the computer science curriculum (e.g. the capacity to implement, test and observe a particular algorithm).

## Course Objectives (Post-conditions)

### Knowledge objectives:

At the conclusion of the course, following learning objectives are expected to be achieved:

1. Explain what constitutes an object-oriented approach to programming and identify potential benefits of Object-oriented programming over other approaches.
2. Analyze and decompose problem specifications from Object Oriented Perspectives and represent the solution, using UML notation.
3. Explain the benefits of object oriented design and the types of systems in which it is an appropriate methodology.
4. Apply an object-oriented approach to developing applications of varying complexities.
5. Augment a class definition using constructors, destructors, member functions, helper functions and custom input/output operators to add functionality to a programming solution
6. Manage an object's resources using dynamic memory allocation and deallocation to access data stored outside the object's memory
7. Read from and write to files using objects from the standard input output library and custom file operators for future restoration
8. Model specialization using single inheritance and abstract base classes to minimize code duplication
9. Model polymorphic behavior using coercion, overloading, virtual functions and function templates to amplify reusability of code

### Application objectives:

The lab work and homework portions of the course are intended to help you apply your understanding,

- Basic programming techniques
- Design object oriented solutions for small systems involving multiple objects.
- Apply good programming style and understand the impact of style on developing and maintaining programs. Be able to justify programming style choices.

- Explain the steps in creating an executable program for a computer, including the intermediate representations and their purpose
- Trace the execution of program code to debug an application

**Expected Student Background (Preconditions)**

Students are expected to have a solid grasp of the introduction to computer programming.

**Topics Outline:**

S NO	Topics	Hrs
1	Review of Structured programming in C, Structured versus O-O programming, O-O paradigm	2
2	Objects, Classes, Methods, Constructors and destroying objects in C++	3
3	Object-oriented Analysis –Class diagrams and Object Diagrams	3
4	File Handling in C++ and JAVA	3
5	Friend Functions, Static member functions in C++ and JAVA	2
6	Complex Objects and Classes	3
7	Inheritance, Multiple Inheritance in C++	3
8	Polymorphism and Virtual Functions in C++	2
9	RTTI and Casting types in C++	2
10	Function and Operator overloading in C++	2
11	Namespace and Templates in C++	2
12	STL-Container classes, Sequence, Iterators	2
13	Java GUI and AWT	3
14	Introduction to Swing and Applet	4
15	Arrays	1
16	Inheritance and Abstract Classes in JAVA	2
17	Packages, Class path, Interfaces in JAVA	2
18	Basic OO Testing and Exception Handling in C++ and JAVA	2
	Total	41

**References**

1. Lafore R., Object oriented programming in C++, Waite Group
2. Java 2: The Complete Reference, Fifth Edition -- by Herbert Schildt
3. Deitel and Deitel How to Program C++
4. Stroustrup B., The C++ Programming Language, Addison Wesley
5. Bruce Eckel, Thinking in Java
6. Java Swings by Robert Eckstein, Marc Loy & Dave Wood, Orelly's

**Evaluation Scheme:**

S.No	Examination	Marks
1	T-1	15
2	T-2	25
3	T-3	35
4	*Internal Marks	25

\*Internal Marks Breakdown:

Assignments            9 marks (3x3)

Quizzes                12 marks (3x4)

Regularity            4 Marks