

# Lab Manual

**CSC—241**

**Object Oriented Programming**



**Department of Computer Science  
Islamabad Campus**

**CUI**

**Lab Contents:**

This lab emphasizes the concepts of object-oriented techniques used in developing computer-based system. The topics include: Problem solving in Object Oriented Paradigm, Defining classes and objects in JAVA, Controlling access to members- Encapsulation, Passing and returning non-primitive values from methods, Composition/Containership (Has-a relationship), Inheritance, Method Overriding and Abstract Class, Polymorphism, Interfaces, Array List and Generics, File Handling, Graphical User Interface – Layout Managers , Graphical User Interface- Event Driven Programming.

**Student Outcomes (SO)**

S.#	Description
1	Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements
2	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines
3	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations
4	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations
5	Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings.
9	Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional

**Intended Learning Outcomes**

Sr.#	Description	Blooms Taxonomy Learning Level	SO
CLO -4	Implement a small module utilizing Object-Oriented design.	<i>Applying</i>	2-4
CLO-5	Develop a GUI based project for a real-world problem in a team environment.	<i>Creating</i>	2-5,9

**Lab Assessment Policy**

The lab work done by the student is evaluated using Psycho-motor rubrics defined by the course instructor, viva-voce, project work/performance. Marks distribution is as follows:

Assignments	Lab Mid Term Exam	Lab Terminal Exam	Total
25	25	50	100

**Note: Midterm and Final term exams must be computer based.**

## List of Labs

<b>Lab #</b>	<b>Main Topic</b>	<b>Page #</b>
Lab 01	Problem Solving in Object Oriented Paradigm	<b>4</b>
Lab 02	Defining classes and objects in JAVA	<b>10</b>
Lab 03	Controlling access to class members – Encapsulation	<b>17</b>
Lab 04	Passing and returning non primitive values from methods	<b>26</b>
Lab 05	Composition / Containership (Has-a relationship)	<b>33</b>
Lab 06	Inheritance	<b>43</b>
Lab 07	Method Overriding and Abstract class	<b>51</b>
Lab 08	Polymorphism	<b>59</b>
Lab 09	<b>Mid Term Exam</b>	
Lab 10	Interfaces	<b>69</b>
Lab 11	Array Lists and Generics	<b>80</b>
Lab 12	File Handling	<b>90</b>
Lab 13	Graphical User Interface - Layout Managers	<b>96</b>
Lab 14	Graphical User Interface – Event Driven Programming	<b>108</b>
Lab 15	<b>Final Term Exam</b>	

# **Lab 01**

## **Problem Solving in Object Oriented Paradigm**

### **Objective:**

Objective of this lab is to understand the Object-Oriented paradigm.

### **Activity Outcomes:**

The student will be able to understand the Object-oriented paradigm.

The student will be able to understand difference between class and object.

### **Instructor Note:**

As pre-lab activity, read Chapter 9 from the text book “Introduction to Java Programming”, Y. Daniel Liang, Pearson, 2019.

## 1) Useful Concepts

The world around us is made up of objects, such as people, automobiles, buildings, streets, and so forth. Each of these objects has the ability to perform certain actions, and each of these actions has some effect on some of the other objects in the world.

OOP is a programming methodology that views a program as similarly consisting of objects that interact with each other by means of actions.

Object-oriented programming has its own specialized terminology. The objects are called, appropriately enough, objects. The actions that an object can take are called methods. Objects of the same kind are said to have the same type or, more often, are said to be in the same class.

For example, in an airport simulation program, all the simulated airplanes might belong to the same class, probably called the Airplane class. All objects within a class have the same methods. Thus, in a simulation program, all airplanes have the same methods (or possible actions), such as taking off, flying to a specific location, landing, and so forth. However, all simulated airplanes are not identical. They can have different characteristics, which are indicated in the program by associating different data (that is, some different information) with each particular airplane object. For example, the data associated with an airplane object might be two numbers for its speed and altitude.

Things that are called procedures, methods, functions, or subprograms in other languages are all called methods in Java. In Java, all methods (and for that matter, any programming constructs whatsoever) are part of a class.

**Syntax :**

```
class ClassName {  
    //datefileds  
    //methods  
}
```

## 2) Solved Lab Activities

<i>Sr.No</i>	<i>Allocated Time</i>	<i>Level of Complexity</i>	<i>CLO Mapping</i>
<i>Activity 1</i>	<i>20 mins</i>	<i>Low</i>	<i>CLO-4</i>
<i>Activity 2</i>	<i>20 mins</i>	<i>Low</i>	<i>CLO-4</i>
<i>Activity 3</i>	<i>20 mins</i>	<i>Low</i>	<i>CLO-4</i>

## Activity 1:

*Consider the concept of a CourseResult. The CourseResult should have data members like the student's name, course name and grade obtained in that course.*

*This concept can be represented in a class as follows:*

## Solution:

```
class CourseResult {

    public String studentname;
    public String coursename;
    public String grade;

    public void display() {
        System.out.println("Student Name is:" +
studentname + "Course Name is:" + coursename
        + "Grade is:" + grade);
    }
}

public class CourseResultRun {

    public static void main(String[] args) {
        CourseResult c1 = new CourseResult();
        c1.studentname = "Ali";
        c1.coursename = "OOP";
        c1.grade = "A";
        c1.display();
        CourseResult c2 = new CourseResult();
        c2.studentname = "Saba";
        c2.coursename = "ICP";
        c2.grade = "A+";
        c2.display();
    }
}
```

Note that both objects; c1 and c2 have three data members, but each object has different values for their data members.

## Activity 2:

*The example below represents a Date class. As date is composed of three attributes, namely month, year and day; so, the class contains three Data Members. Now every date object will have these three attributes, but each object can have different values for these three*

## Solution:

```
class Date {

    public String month;
    public int day;
    public int year; //a four digit number.

    public void displayDate() {
        System.out.println(month + " " + day + ", " + year);
    }
}

public class DateDemo {

    public static void main(String[] args) {
        Date date1, date2;
        date1 = new Date();
        date1.month = "December";
        date1.day = 31;
        date1.year = 2012;
        System.out.println("date1:");
        date1.displayDate();
        date2 = new Date();
        date2.month = "July";
        date2.day = 4;
        date2.year = 1776;
        System.out.println("date2:");
        date2.displayDate();
    }
}
```

### Activity 3:

*Consider the concept of a Car Part. After analyzing this concept we may consider that it can be described by three data members: modelNumber, partNumber and cost.*

*The methods should facilitate the user to assign values to these data members and show the values for each object.*

*This concept can be represented in a class as follows:*

```
import java.util.Scanner;

class CarPart {

    private String modelNumber;
    private String partNumber;
    private String cost;

    public void setparameter(String x, String y, String z) {
        modelNumber = x;
        partNumber = y;
        cost = z;
    }

    public void display() {
        System.out.println("Model Number: " + modelNumber + "Part
Number: " + partNumber+ "Cost: " + cost);
    }
}

public class CarPartRunner {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        CarPart car1 = new CarPart();
        System.out.println("What is Model Number?");
        System.out.println("What is Part Number?");
        System.out.println("What is Cost?");
        String x = sc.nextLine();
        String y = sc.nextLine();
        String z = sc.nextLine();
        car1.setparameter(x, y, z);
        car1.display();
    }
}
```



```
}  
}
```

### 3) Graded Lab Tasks

*Note: The instructor can design graded lab activities according to the level of difficulty and complexity of the solved lab activities. The lab tasks assigned by the instructor should be evaluated in the same lab.*

#### Lab Task 1

*A Student is an object in a university management System. Analyze the concept and identify the data members that a Student class should have. Also analyze the behavior of student in a university management System and identify the methods that should be included in Student class.*

#### Lab Task 2

*Time is an intangible concept. Analyze the concept and identify the data members and methods that should be included in Time class.*

#### Lab Task 3

*Car is an object that helps us in transportation. Analyze the concept and identify the data members and methods that should be included in Car class.*

#### Lab Task 4

*Rectangle is an object that represents a specific shape. Analyze the concept and identify the data members and methods that should be included in Rectangle class.*