OBject oriented programming with java

Lab manual

Aymen Khan

Aymkhan498@gmail.com

**Course Title:** Object Oriented Programming

**Course Code:**

**Course Outcomes:**

At the end of the course the student should be able to:

1. Apply Object Oriented Programming concepts to solve a given problem.
2. Apply design patterns to design a solution for a given problem.
3. Apply inheritance, polymorphism and exception handling mechanism to implement reusable, robust java programs.
4. Implement user interface java programs for a given scenario.

**List of Practical**

|  |  |  |
| --- | --- | --- |
| **Sr.#** | **Week** | **Topics** |
| 1 | One | Introduction to IDE-NetBeans, Getting Started with Java |
| 2 | Two | Java Basics (Input/output, variable declaration and initialization, strings and arrays) |
| 3 | Three | Java Basics (Selection Structure and Iterative Structure, functions) |
| 4 | Four | OOP (creating classes, objects, constructors) |
| 5 | Five | access modifiers, inheritance |
| 6 | Six | multiple/multilevel inheritance |
| 7 | Seven | Function overriding |
| 8 | Eight | Polymorphism |
| 9 | Nine | abstract classes and interfaces |
| 10 | Ten | exception handling and Java file handling |

**Introduction to NetBeans and Java**

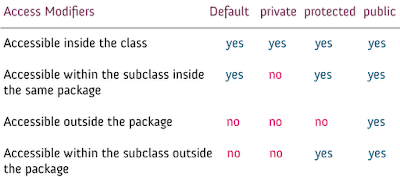
**LAB-1**

**OOP (access modifiers, inheritance)**

**LAB-5**

**Access Modifiers**

Specifiers determine whether a field or method in a class, can be used or invoked by another method in another class or sub-class. Access Specifiers can be used to restrict access.



**Example:**

public class Person { // public class

private int age; // private (encapsulated) instance variables

public set(int ag) { // setting values of private fields

this.age = x;

}

public get() { // setting values of private fields

return Point(age);

}

}

**Inheritance**

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance represents the IS-A relationship which is also known as a parent-child relationship.

class Subclass-name extends Superclass-name

{

//methods and fields

}

e.g.

Create a class SHAP that have two data members height and width and two member functions input () and output () input take height and width from the user and output display the height and width. And make another class RECTANGLE that inherit from the SHAP that use the data members and function of the SHAP class. In main make two RECTANGLE objects and use using input and output function.

ANS:

private class shape{

Protected float height,width;

public shap(){height=0.0;width=0.0;}

public get(){

cout<<"Enter Height\n";

cin>>height;

cout<<"Enter Width\n";

cin>>width;}

public void show(){cout<<"Height : "<<height<<"\nWidth : "<<width;

}

}

public class rectangle extends shape{

};

int main(){

rectangle r;

r.get();

r.show();

getch();

}

**Task-1**

Using above example take another class COST. RECTANGLE that inherit publically also COST. In COST one member function that calculate the cost according to the area. In main create two object of RECTANGLE class in one object initialize height and width and in second take from the user and calculate the area. And area is a member function of RECTANGLE class after calculating the area the area is send to the COST class that calculate the area and display it.

**Task-2**

A point in the x-y plane is represented by its x-coordinate and y-coordinate. Design a class, point Type, that can store and process a point in the x-y plane. You should then perform operations on the point, such as setting the coordinates of the point, printing the coordinates of the point, returning the x-coordinate, and returning the y-coordinate. Also, write a program to test various operations on the point.

**Assignment Question**

Every circle has a center and a radius. Given the radius, we can determine the circle’s area and circumference. Given the center, we can determine its position in the x-y plane. The center of the circle is a point in the x-y plane. Design a class, circleType, that can store the radius and center of the circle. Because the center is a point in the x-y plane and you designed the class to capture the properties of a point in Programming task 2, you must derive the class circleType from the class pointType. You should be able to perform the usual operations on the circle, such as setting the radius, printing the radius, calculating and printing the area and circumference, and carrying out the usual operations on the center. Also, write a program to test various operations on a circle.