EXPERIMENT 06

Aim of the Experiment- To demonstrate the implementation and practical application of polymorphism and inheritance in Java.

Problem 6.1: A class 'B0x ' which contains methods and other details(width,height,depth) to calculate volume . Another class 'BoxWight' contains methods for finding cost for shipping to box using weight of the box(formula to find cost of shipping = distance in km * volume * cost per km) Take cost per KM by the user. WAP to find the volume of the box and cost to ship using a single object

Solution 6.1:

```
package ex6;

import java.util.Scanner;

class Box {

    double width;

    double height;

    double depth;

    Box(double w, double h, double d) {

        width = w;

        height = h;

        depth = d;}

    double volume() {

        return width * height * depth;}}

class BoxWeight extends Box {

    double distance;

    double costPerKm;

    BoxWeight(double w, double h, double d, double dist, double cost) {

        super(w, h, d);

        distance = dist;

        costPerKm = cost;}

    double shippingCost() {

        return distance * volume() * costPerKm;}

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        double w = sc.nextDouble();

        double h = sc.nextDouble();

        double d = sc.nextDouble();

        double dist = sc.nextDouble();

        double cost = sc.nextDouble();

        sc.close();
```

```
        BoxWeight bw = new BoxWeight(w, h, d, dist, cost);
        System.out.println("Box Volume: "+ bw.volume());
        System.out.println("Shipping Cost: "+ bw.shippingCost());}}
```

Output 6.1:

```
5
6
3
20
25
Box Volume: 90.0
Shipping Cost: 45000.0
```

Problem 6.2: Shape is the super class for square and rectangle.write a program to find areas of square and rectangle by inheriting Method-find the area() should be declared in shape .parameters could changed as per needed

Solution 6.2:

```
package ex6;
class Shape {
    double area() {
        return 0;}}
class Square extends Shape {
    double side;
    Square(double s) {
        side = s;}
    double area() {
        return side * side;}}
class Rectangle extends Shape {
    double length;
    double breadth;
    Rectangle(double l, double b) {
        length = l;
        breadth = b;}
    double area() {
        return length * breadth;}
    public static void main(String[] args) {
        Shape s1 = new Square(5);
        Shape s2 = new Rectangle(4, 6);
        System.out.println("Area of Square: "+ s1.area());
        System.out.println("Area of Rectangle: "+ s2.area());}}
```

Output 6.2:

```
Area of Square: 25.0
Area of Rectangle: 24.0
```

Problem 6.3: Demonstrate constructor overloading with an example of employee class to print different details of an employee based on the parameters passed Demonstration of Constructor Overloading

Solution 6.3:

```
package ex6;

class Employee {
   int id;
   String name;
   double salary;
   Employee(int i) {
      id = i;}
   Employee(int i, String n) {
      id = i;
      name = n;}
   Employee(int i, String n, double s) {
      id = i;
      name = n;
      salary = s;}
   void display() {
      System.out.println("ID: "+id);
      System.out.println("Name: "+name);
      System.out.println("Salary: "+salary);}
   public static void main(String[] args) {
      Employee e1 = new Employee(101);
      Employee e2 = new Employee(102, "Amit");
      Employee e3 = new Employee(103, "Riya", 50000);
      e1.display();
      e2.display();
      e3.display();}}
```

Output 6.3:

```
ID: 101
Name: null
Salary: 0.0
ID: 102
Name: Amit
Salary: 0.0
ID: 103
Name: Riya
Salary: 50000.0
```

Problem 6.4: Class shape contains a method 'about' . its subclass 'circle' also has method 'about' but different body. is it possible to create an object of a circle but give its reference as shape? demonstrate via example with suitable output

Solution 6.4:

```
package ex5;

interface Shape {
    double about();}

class Circle implements Shape {
    double r;
    Circle(double r) {
        this.r = r;}
    public double about() {
        return 3.14 * r * r;}}

class Rectangle implements Shape {
    double a, b;
    Rectangle(double a, double b) {
        this.a = a;
        this.b = b;}
    public double about() {
        return a * b;}}

class Main {
    public static void main(String[] args) {
        Shape c = new Circle(5);
        Shape r = new Rectangle(5, 3);
        System.out.println("Area of Circle is: " + c.about());
        System.out.println("Area of Rectangle is: " + r.about());}}
```

Output 6.4:

```
Area of Circle is: 78.5
Area of Rectangle is: 15.0
```

--Submitted By--
Name-Jitesh Kumar Das
Regd_no-2401020092
Branch-CSE
Sem-4th
Gr-01