

OBJECT ORIENTED PROGRAMMING LAB

EXERCISE SET 5

ABSTRACT CLASSES AND POLYMORPHISM

1. Redo the Exercise 6 of the previous set after changing the Figure class to an abstract class with abstract area() method and calling the overridden methods through a superclass reference.

```
abstract class Figure {
    private double dim1;
    private double dim2;

    public Figure(double a, double b) {
        dim1 = a;
        dim2 = b;
    }
    abstract double area();//abstract methods have no body

    final double getDim1(){return dim1;}
    final double getDim2(){return dim2;}
}
```

2. Consider the abstract Shape class defined as follows:

```
abstract class Shape{
    double area(){return 0;}
    double volume(){return 0;}

    abstract void printShapeName();//abstract methods have no body
    abstract void print();//abstract methods have no body
}
```

Extend the Shape class to Point class that describes a point in 2D space. Add two data members x and y of double type. Override the abstract methods in Shape as follows:

```
void printShapeName(){
    System.out.print("\nPoint\t");
}
void print(){
    System.out.println "[" + x + "," + y + "]";
}
```

Now, extend the Point class to Circle class that describes a circle in 2D space. Add a data member radius of double type. Override the printShapeName() and print() methods again suitably. Also, add a method to Circle class that computes the area of the circle.

Finally, extend the Circle to Cylinder class. Implement a driver class to demonstrate the various overridden method through a superclass reference.

Redo the above exercise by adding an Ellipse class to above class hierarchy.