1. Create a class Shape having atmost two dimensions. Define two subclasses circle and rectan gle of Shape.

i) Override a method area.

ii)Show compile time and run time polymorphism(Dynamic method dispatch).

iii)Use a final method for display.

iv) use super keyword.

CODE :-

```
// Base class
class Shape {
    double dim1, dim2;

    // Constructor
    Shape(double d1, double d2) {
        this.dim1 = d1;
        this.dim2 = d2;
    }

    // Default area method (can be overridden)
    double area() {
        System.out.println("Area of Shape is undefined");
        return 0;
    }

    // Final method to display area - cannot be overridden
    final void display() {
        System.out.println("Displaying area using final method.");
    }

    // Compile-time polymorphism: Overloaded method to set dimensions
    void setDimensions(double d1) {
        this.dim1 = d1;
        this.dim2 = 0;
    }
```

```
    void setDimensions(double d1, double d2) {
        this.dim1 = d1;
        this.dim2 = d2;
    }
}

// Subclass Circle
class Circle extends Shape {
    Circle(double radius) {
        super(radius, 0);  // super keyword to call parent constructor
    }

    // Override area method (runtime polymorphism)
    @Override
    double area() {
        return Math.PI * dim1 * dim1; // dim1 as radius
    }

    void displayCircle() {
        super.display(); // calling final method of super class
        System.out.println("Area of Circle with radius " + dim1 + " = " + area());
    }
}

// Subclass Rectangle
class Rectangle extends Shape {
    Rectangle(double length, double breadth) {
        super(length, breadth);  // super keyword to call parent constructor
    }

    // Override area method (runtime polymorphism)
```

```
    @Override
    double area() {
        return dim1 * dim2;
    }

    void displayRectangle() {
        super.display(); // calling final method of super class
        System.out.println("Area of Rectangle with length " + dim1 + " and breadth " + dim2 + " = " + area());
    }
}

// Main class to test
public class TestShape {
    public static void main(String[] args) {
        // Compile time polymorphism - method overloading
        Shape shape = new Shape(0, 0);
        shape.setDimensions(5);  // sets dim1=5, dim2=0
        System.out.println("Shape dimensions set to one parameter: dim1 = " + shape.dim1 + ", dim2 = " + shape.dim2);

        shape.setDimensions(5, 10); // sets dim1=5, dim2=10
        System.out.println("Shape dimensions set to two parameters: dim1 = " + shape.dim1 + ", dim2 = " + shape.dim2);

        // Runtime polymorphism - dynamic method dispatch
        Shape s;

        s = new Circle(7);
        System.out.println("Area of Circle: " + s.area());  // Circle's area()

        s = new Rectangle(4, 8);
        System.out.println("Area of Rectangle: " + s.area());  // Rectangle's area()
```

```
        // Demonstrate final method and super keyword
        Circle c = new Circle(3);
        c.displayCircle();

        Rectangle r = new Rectangle(6, 2);
        r.displayRectangle();
    }
}
```

Output :-