Name: Somnath R. Shintre Roll No:

Class: TE CSE Batch:

**Title: -** Create abstract class Shape which has instance variables side, area and perimeter and methods calculateArea(), calculatePerimeter() as abstract methods and display() as concrete method. Write subclasses which extend Shape class like Triangle, Rectangle, Circle, Cube and Square and override abstract methods and display methods in subclass take instance variable if needed as per the formula. And use parameterized constructor to initialize instance variables using “this” reference variable. Write Test class and Create a reference variable of Shape which will hold the objects of all the sub classes and calculate respective area, perimeter and display the results.

**Program:-**

/\*\*

 \* The Shape class is an abstract class that holds general data about a shape.

 \* Super class!!

 \*/

abstract class Shape {

    // Instance variables

    int side;

    double area;

    double perimeter;

    // abstract methods

    // It must be overridden in a subclass.

    abstract void calculateArea();

    abstract void calculatePerimeter();

    // concrete method

    public void display() {

        System.out.println("The area is " + area);

        System.out.println("The perimeter is " + perimeter);

    }

}

/\*\*

 \* This class Triangle is extended from shape abstaract class.

 \*/

class Triangle extends Shape {

    // Instance variables

    // side variable from the Shape abstract class holds the base lenght of the

    // triangle

    int height;

    // Parameterized Constructor

    public Triangle(int side, int height) {

        this.side = side;

        this.height = height;

    }

    // Overiding the abstract calculateArea method from Shape abstract class

    @Override

    void calculateArea() {

        area = 0.5 \* side \* height;

    }

    // Overiding the abstract calculatePerimeter method from Shape abstract class

    @Override

    void calculatePerimeter() {

        perimeter = side + height + (Math.pow(Math.pow(side, 2) + Math.pow(height, 2), 0.5));

    }

}

/\*\*

 \* This class Rectangle is extended from shape abstaract class

 \*/

class Rectangle extends Shape {

    // Instance variables

    // side variable from the Shape abstract class holds the lenght of the rectangle

    int breadth;

    // Parameterized Constructor

    public Rectangle(int side, int breadth) {

        this.side = side;

        this.breadth = breadth;

    }

    // Overiding the abstract calculateArea method from Shape abstract class

    @Override

    void calculateArea() {

        area = side \* breadth;

    }

    // Overiding the abstract calculatePerimeter method from Shape abstract class

    @Override

    void calculatePerimeter() {

        perimeter = 2 \* (side + breadth);

    }

}

/\*\*

 \* This class Circle is extended from shape abstaract class.

 \*/

class Circle extends Shape {

    // side variable from the Shape abstract class holds the radius of the circle

    // Parameterized Constructor

    public Circle(int side) {

        this.side = side;

    }

    // Overiding the abstract calculateArea method from Shape abstract class

    @Override

    void calculateArea() {

        area = 3.1415 \* side \* side;

    }

    // Overiding the abstract calculatePerimeter method from Shape abstract class

    @Override

    void calculatePerimeter() {

        perimeter = 2 + 3.1415 \* side;

    }

}

/\*\*

 \* This class Cube is extended from shape abstaract class.

 \*/

class Cube extends Shape {

    // side variable from the Shape abstract class holds the side length of the cube

    // Parameterized Constructor

    public Cube(int side) {

        this.side = side;

    }

    // Overiding the abstract calculateArea method from Shape abstract class

    @Override

    void calculateArea() {

        area = 6 \* (side \* side);

    }

    // Overiding the abstract calculatePerimeter method from Shape abstract class

    @Override

    void calculatePerimeter() {

        perimeter = 12 \* side;

    }

}

/\*\*

 \* This class Squere is extended from shape abstaract class.

 \*/

class Square extends Shape {

    // side variable from the Shape abstract class holds the side length of the

    // square

    // Parameterized Constructor

    public Square(int side) {

        this.side = side;

    }

    // Overiding the abstract calculateArea method from Shape abstract class

    @Override

    void calculateArea() {

        area = side \* side;

    }

    // Overiding the abstract calculatePerimeter method from Shape abstract class

    @Override

    void calculatePerimeter() {

        perimeter = 4 \* side;

    }

}

public class Shape\_Abstract {

    public static void main(String[] args) {

        // Creating objects or instances

        // Creating a reference variable of Shape which will hold the objects of all the

        // subclasses.

        // Down casting!

        Shape triangle = new Triangle(15, 20);

        Shape rectangle = new Rectangle(20, 30);

        Shape circle = new Circle(30);

        Shape cube = new Cube(40);

        Shape square = new Square(45);

        System.out.println("\nShapes and there respective area and perimeter\n");

        // Calculating area and perimeter of triangle and displaying

        System.out.println("\n---------- The Triangle shape -----------");

        triangle.calculateArea();

        triangle.calculatePerimeter();

        triangle.display();

        // Calculating area and perimeter of rectangle and displaying

        System.out.println("\n---------- The Rectangle shape ----------");

        rectangle.calculateArea();

        rectangle.calculatePerimeter();

        rectangle.display();

        // Calculating area and perimeter of circle and displaying

        System.out.println("\n---------- The Circle shape -------------");

        circle.calculateArea();

        circle.calculatePerimeter();

        circle.display();

        // Calculating area and perimeter of cube and displaying

        System.out.println("\n---------- The Cube shape ---------------");

        cube.calculateArea();

        cube.calculatePerimeter();

        cube.display();

        // Calculating area and perimeter of square and displaying

        System.out.println("\n---------- The Square shape -------------");

        square.calculateArea();

        square.calculatePerimeter();

        square.display();

    }

}

**Output:-**

