

Exercise 4

Hariesh R - 23110344

Aim:

The program illustrates key Java concepts: defining an abstract class Shape with a method area() and implementing it in Rectangle, Triangle, and Circle to print their areas. It also covers method overriding by providing specific implementations in subclasses and method overloading by using the same method name with different parameters.

Algorithm:

1. i) Shape.java

- Define an abstract class Shape.
- Declare two protected integers integer1 and integer2.
- Initialize these integers via a constructor.
- Declare an abstract method area() for calculating the shape's area.

ii) Rectangle.java

- Define class Rectangle extending Shape.
- Implement constructor to accept width and height, and pass them to the superclass constructor.
- Override area() method to calculate and print the rectangle's area using width * height.

iii) Triangle.java

- Define class Triangle extending Shape.
- Implement constructor to accept base and height, and pass them to the superclass constructor.
- Override area() method to calculate and print the triangle's area using $0.5 * \text{base} * \text{height}$.

iv) Circle.java

- Define class Circle extending Shape.
- Declare constant PI for π .
- Implement constructor to accept radius and pass it to the superclass constructor.
- Override area() method to calculate and print the circle's area using $\pi * \text{radius}^2$.

v) Main.java

- Create instances of Rectangle, Circle, and Triangle using Shape references.
- Call the area() method on each shape instance to print the area of each shape.

2. Main.java

- Define class Animal with method makeSound() printing "Animal is making sound".
- Define class Dog extending Animal, override makeSound() to print "Dog is making sound".
- Define class Cat extending Animal, override makeSound() to print "Cat is making sound".
- In the Main class:
 - Create instances of Animal, Dog, and Cat.
 - Call makeSound() on each instance to demonstrate method overriding.

3. Main.java

- Define class Calculator with multiple add() methods:
 - add(int a, int b): Adds two integers.
 - add(int a, int b, int c): Adds three integers.
 - add(double a, double b): Adds two doubles.
 - add(int[] numbers): Adds elements of an integer array.
- In the Main class:
 - Create an instance of Calculator.
 - Demonstrate method overloading by calling each version of add() with different parameters and print the results.

Code:

1. i) Shape.java

```
package Exercise4.Q1;

public abstract class Shape {

    protected int integer1;
    protected int integer2;

    public Shape(int integer1, int integer2){
        this.integer1 = integer1;
        this.integer2 = integer2;
    }

    abstract void area();
}
```

ii) Rectangle.java

```
package Exercise4.Q1;

public class Rectangle extends Shape{
    public Rectangle(int width, int height){
        super(width, height);
    }
    @Override
    public void area(){

        int area = integer1 * integer2;
        System.out.println("Area of Rectangle: " + area);
    }
}
```

iii) Triangle.java

```
package Exercise4.Q1;
```

```

public class Triangle extends Shape{
    public Triangle(int base, int height){
        super(base, height);
    }

    @Override
    public void area(){

        double area = integer1 * integer2 * 0.5;
        System.out.println("Area of Triangle is: " + area);
    }
}

```

iv) Circle.java

```

package Exercise4.Q1;

```

```

public class Circle extends Shape{

    public static final double PI = 3.14159;

    public Circle(int radius){

        super(radius, 0);
    }

    @Override
    public void area(){

        double area = integer1 * integer1 * PI;
        System.out.println("Area of Circle: " + area);
    }
}

```

v) Main.java

```

package Exercise4.Q1;

```

```

public class Main {

```

```

        public static void main(String[] args) {
            Shape rectangle = new Rectangle(5, 10);
            Shape circle = new Circle(5);
            Shape triangle = new Triangle(5, 15);
            rectangle.area();
            circle.area();
            triangle.area();
        }
    }
}

```

2. Main.java

```

package Exercise4.Q2;

class Animal{

    void makeSound(){

        System.out.println("Animals is making sound");
    }
}

class Dog extends Animal{

    @Override
    void makeSound(){

        System.out.println("Dog is making sound");
    }
}

class Cat extends Animal{

    @Override
    void makeSound(){
        System.out.println("Cat is making sound");
    }
}

```

```

public class Main {
    public static void main(String[] args) {
        Animal animal = new Animal();
        Animal dog = new Dog();
        Animal cat = new Cat();

        animal.makeSound();
        dog.makeSound();
        cat.makeSound();
    }
}

```

3. Main.java

```

package Exercise4.Q3;

class Calculator{

    int add(int a, int b){
        return a + b;
    }

    int add(int a, int b, int c){
        return a + b + c;
    }

    double add(double a, double b){

        return a + b;
    }
    int add(int[] numbers){

        int sum = 0;

        for(int i:numbers){

            sum += i;
        }
    }
}

```

```

        return sum;
    }
}

public class Main {
    public static void main(String[] args) {

        Calculator calc = new Calculator();

        System.out.println("Sum of 5 and 10: " + calc.add(5, 10));
        System.out.println("Sum of 5, 10, and 15: " + calc.add(5, 10, 15));
        System.out.println("Sum of 5.5 and 10.5: " + calc.add(5.5, 10.5));
        System.out.println("Sum of array elements: " + calc.add(new int[]{1, 2, 3,
4, 5}));
    }
}

```

Output:

1.

```

snu-cse@snu-cse-HP-ProDesk-400-G7-Microtower-PC:~/harizzesh$ java Exercise4.Q1.Main
Area of Rectangle: 50
Area of Circle: 78.53975
Area of Triangle is: 37.5
snu-cse@snu-cse-HP-ProDesk-400-G7-Microtower-PC:~/harizzesh$

```

2.

```

snu-cse@snu-cse-HP-ProDesk-400-G7-Microtower-PC:~/harizzesh$ java Exercise4.Q2.Main
Animals is making sound
Dog is making sound
Cat is making sound
snu-cse@snu-cse-HP-ProDesk-400-G7-Microtower-PC:~/harizzesh$

```

3.

```

snu-cse@snu-cse-HP-ProDesk-400-G7-Microtower-PC:~/harizzesh$ java Exercise4.Q3.Main
Sum of 5 and 10: 15
Sum of 5, 10, and 15: 30
Sum of 5.5 and 10.5: 16.0
Sum of array elements: 15
snu-cse@snu-cse-HP-ProDesk-400-G7-Microtower-PC:~/harizzesh$

```

Result:

The program will demonstrate area calculations for different shapes and method behaviors. For the Shape abstract class and its subclasses Rectangle, Triangle, and Circle, the output will show the respective areas of these shapes. The method overriding example will print specific sounds for Animal, Dog, and Cat. The method overloading example will show how different add() methods handle integers, doubles, and integer arrays, displaying sums accordingly.