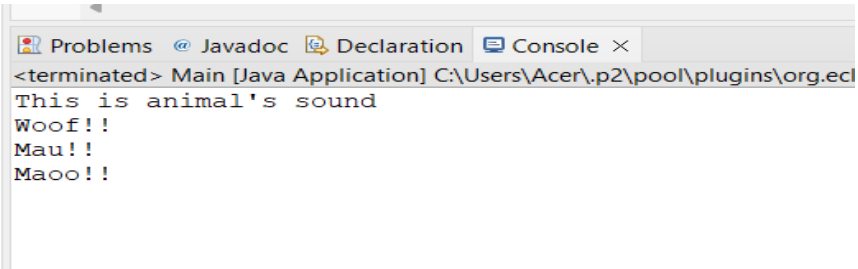
Core_module_java Mock Test-2

Q=>1

Create a superclass called Animal with a method makeSound() that prints the sound made by the animal. Implement subclasses Dog, Cat, and Cow that inherit from the Animal class. Implement the makeSound() method in each subclass to print the sound made by a dog, cat, and cow, respectively.

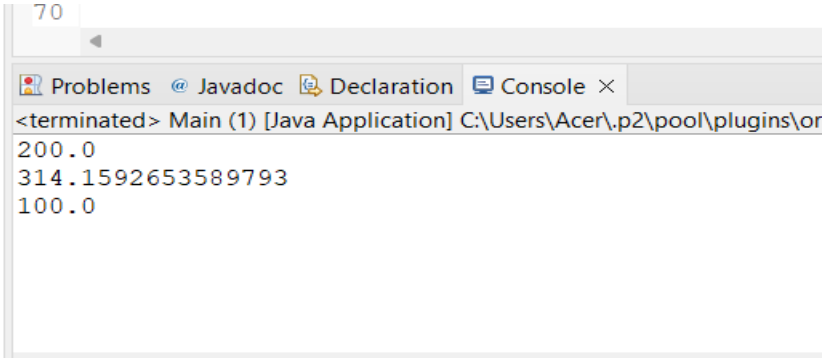
```
package animal_java;


class Animal{
	public void makeSound() {
		System.out.println("This is animal's sound");
	}
}

class Dog extends Animal{
	@override
	public void makeSound() {
		System.out.println("Woof!!");
	}

}
class Cat extends Animal{
	@override
	public void makeSound() {
		System.out.println("Mau!!");
	}
}

class Cow extends Animal{
	@override
	public void makeSound() {
		System.out.println("Maoo!!");
	}
}
public class Main {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Animal animal = new Animal();
		animal.makeSound();

		Dog dog  = new Dog();
		dog.makeSound();

		Cat cat = new Cat();
		cat.makeSound();

		Cow cow = new Cow();
		cow.makeSound();



	}

}
```



Q=>2 Create a superclass called Shape with an abstract method calculateArea() that returns the area of the shape. Implement subclasses Rectangle, Circle, and Triangle that inherit from the Shape class. Implement the calculateArea() method in each subclass to calculate and return the area of a rectangle, circle, and triangle, respectively. Then, create a class called ShapeCalculator with a method printArea(Shape shape) that accepts an object of type Shape and prints its area. Demonstrate polymorphism by passing instances of different subclasses to the printArea() method.

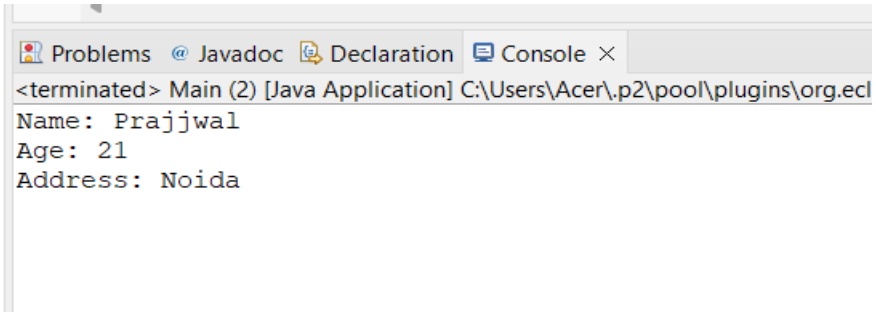
```
package shape;

abstract class Shape{
	abstract double calculateArea();

}

class Rectangle extends Shape{
	private double width;
	private double height;

	public Rectangle(double width, double height) {
		this.width = width;
		this.height = height;
	}
	@Override
	double calculateArea() {
		return width*height;
	}
}

class Circle extends Shape{
	private double radius;

	public Circle(double radius) {
		this.radius = radius;
	}
	@Override
	double calculateArea() {
		return Math.PI*radius*radius;
	}
}
class Triangle extends Shape{
	private double length;
	private double width;

	public Triangle(double length, double width) {
		this.length = length;
		this.width = width;
	}
	@Override
	double calculateArea() {
		return 0.5*length*width;
	}
}

class ShapeCalculator {
	public void printArea(Shape shape) {
		System.out.println(shape.calculateArea());
	}
}


public class Main {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Rectangle rr = new Rectangle(10,20);
```

```
            Circle cc = new Circle(10);
            Triangle tt = new Triangle(10,20);


            ShapeCalculator shapCal = new ShapeCalculator();
            shapCal.printArea(rr);
            shapCal.printArea(cc);
            shapCal.printArea(tt);
      }

}
```




```
package person;

class Person{
      private String name;
      private int age;
      private String Address;

      public void setName(String name) {
            this.name = name;
      }
      public String getName() {
            return name;
      }

      public void setAge(int age) {
            this.age = age;
      }
      public int getAge() {
            return age;
      }

      public void setAddress(String Address) {
            this.Address = Address;
      }
      public String getAddress() {
            return Address;
      }
}

public class Main {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            Person pp = new Person();

            pp.setName("Prajjwal");
            pp.setAge(21);
            pp.setAddress("Noida");
```

```
            System.out.println("Name: " + pp.getName());
            System.out.println("Age: " + pp.getAge());
            System.out.println("Address: " + pp.getAddress());
      }

}
```




```
interface Drawable {
    void draw();
}

class Circle implements Drawable {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public void draw() {
        System.out.println("Drawing a circle with radius " + radius);
    }
}

class Rectangle implements Drawable {
    private double width;
    private double height;

    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    @Override
    public void draw() {
        System.out.println("Drawing a rectangle with width " + width + " and
height " + height);
    }
}

public class Main {
    public static void main(String[] args) {
        Circle circle = new Circle(10);
        Rectangle rectangle = new Rectangle(20, 30);

        circle.draw();
```

```
    rectangle.draw();
}
  }
```