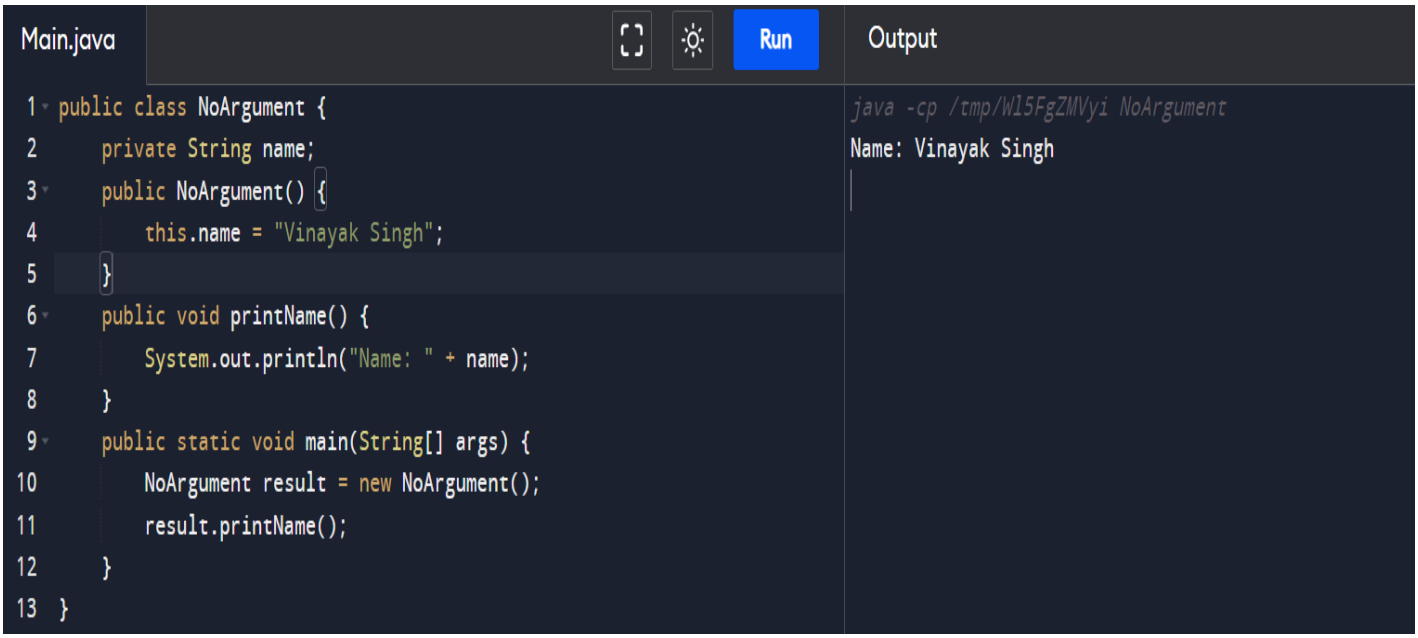
RegNo: 23MCA1030 **Name**: Vinayak Kumar Singh

Java Programming Lab (PMCA502P)

1.Write a Java program to demonstrate a no-argument constructor.

Code:

```
public class NoArgument {
    private String name;
    public NoArgument() {
        this.name = "Vinayak Singh";
    }
    public void printName() {
        System.out.println("Name: " + name);
    }
    public static void main(String[] args) {
        NoArgument result = new NoArgument();
        result.printName();
    }
}
```



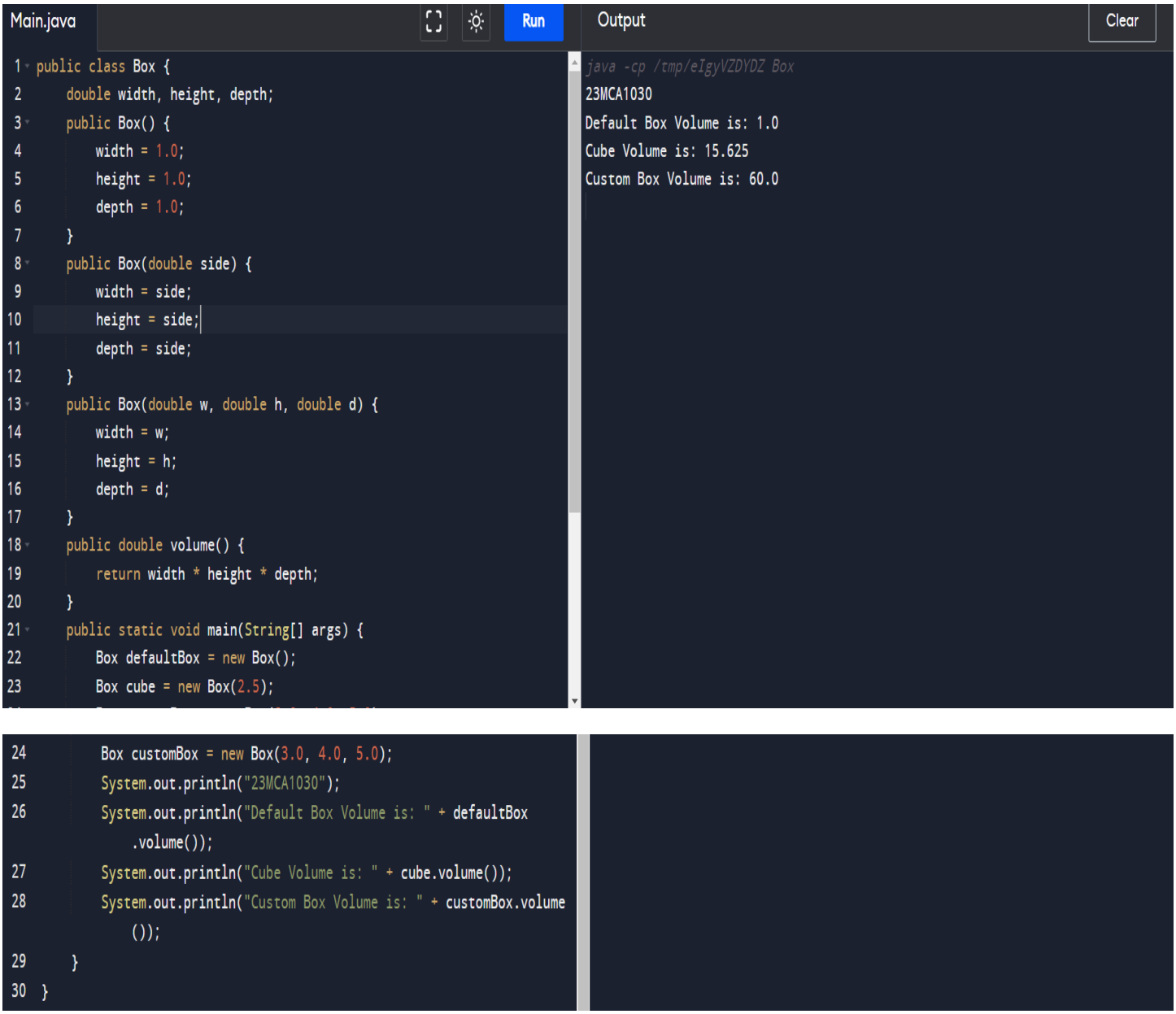
2. Create a program that shows constructor overloading in Java.

Code:

```
public class Box {
    double width, height, depth;
    public Box() {
        width = 1.0;
        height = 1.0;
        depth = 1.0;
    }
    public Box(double side) {
        width = side;
        height = side;
        depth = side;
    }
    public Box(double w, double h, double d) {
        width = w;
        height = h;
        depth = d;
    }
    public double volume() {
        return width * height * depth;
    }
    public static void main(String[] args) {
        Box defaultBox = new Box();
        Box cube = new Box(2.5);
        Box customBox = new Box(3.0, 4.0, 5.0);
        System.out.println("23MCA1030");
```

```
        System.out.println("Default Box Volume is: " + defaultBox.volume());
        System.out.println("Cube Volume is: " + cube.volume());
        System.out.println("Custom Box Volume is: " + customBox.volume());
    }
}
```

Output:



3. Implement a parameterized constructor to initialize an object

Code:

```
public class Student {
    String name;
    int age;
    String course;
    public Student(String studentName, int studentAge, String studentCourse) {
        name = studentName;
        age = studentAge;
        course = studentCourse;
    }
    public void displayStudentInfo() {
        System.out.println("Student Name: " + name);
        System.out.println("Student Age: " + age);
        System.out.println("Student Course: " + course);
    }
    public static void main(String[] args) {
        Student student1 = new Student("Vinayak Singh", 21, "MCA");
        System.out.println("23MCA1030");
        System.out.println("Student Information:");
        student1.displayStudentInfo();
    }
}
```

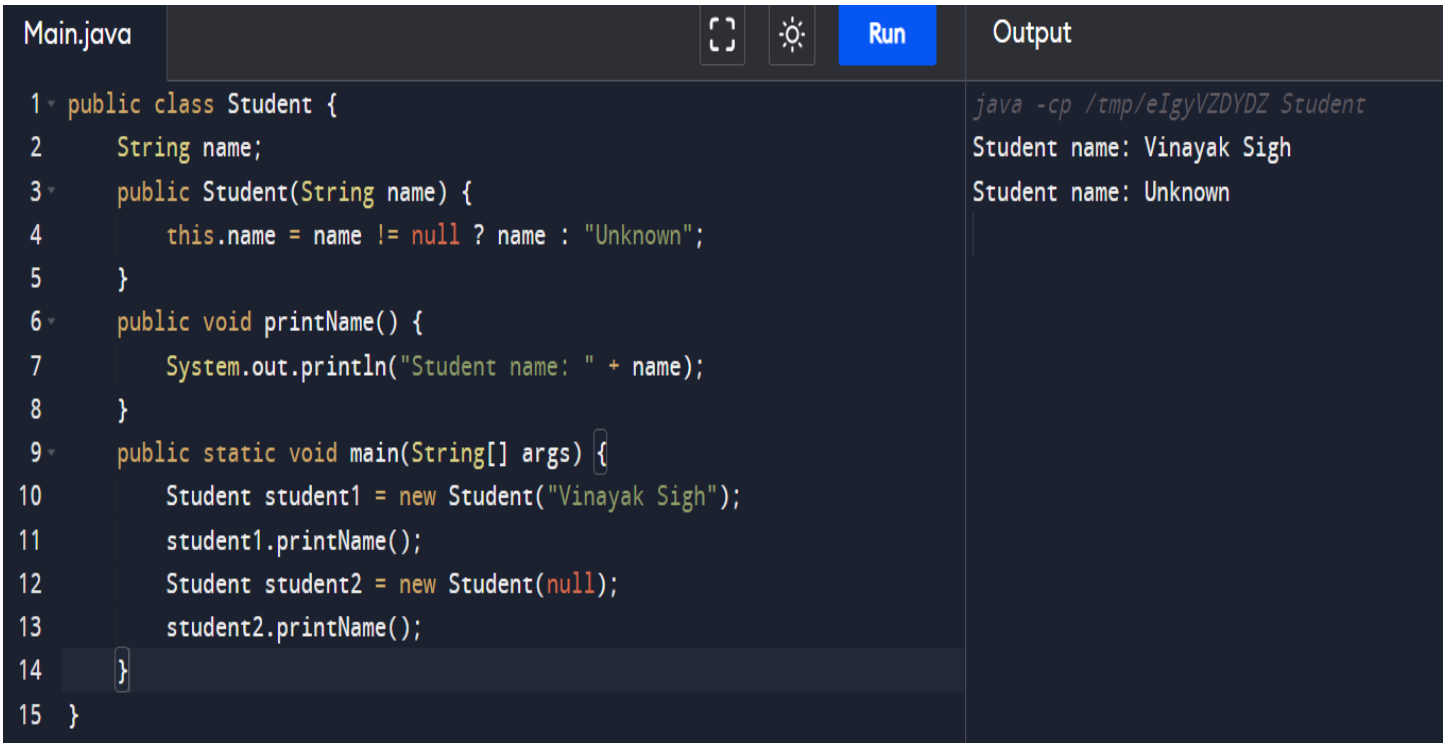
Output:

4. Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating object of Student class.

Code:

```
public class Student {
    String name;
    public Student(String name) {
        this.name = name != null ? name : "Unknown";
    }
    public void printName() {
        System.out.println("Student name: " + name);
    }
    public static void main(String[] args) {
        Student student1 = new Student("Vinayak Sigh");
        student1.printName();
        Student student2 = new Student(null);
        student2.printName();
    }
}
```

Output:



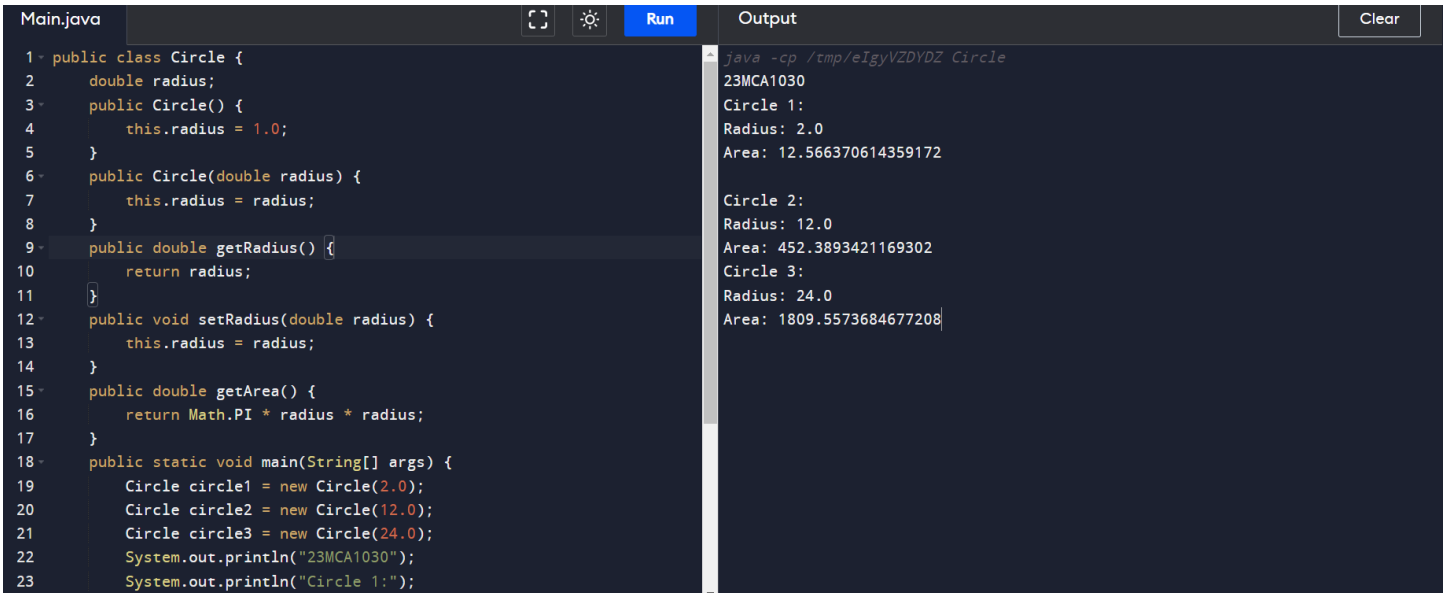
5. Design a class named Circle. Construct three circle objects with radius 2.0, 12, and 24 and displays the radius and area of each. A no-arg constructor set the default value of radius to 1. A getArea() function is used to return the area of circle. Now implement the class.

Code:

```
public class Circle {
    double radius;
    public Circle() {
        this.radius = 1.0;
    }
    public Circle(double radius) {
        this.radius = radius;
    }
    public double getRadius() {
        return radius;
    }
```

```
    public void setRadius(double radius) {
        this.radius = radius;
    }
    public double getArea() {
        return Math.PI * radius * radius;
    }
    public static void main(String[] args) {
        Circle circle1 = new Circle(2.0);
        Circle circle2 = new Circle(12.0);
        Circle circle3 = new Circle(24.0);
        System.out.println("23MCA1030");
        System.out.println("Circle 1:");
        System.out.println("Radius: " + circle1.getRadius());
        System.out.println("Area: " + circle1.getArea());
        System.out.println("\nCircle 2:");
        System.out.println("Radius: " + circle2.getRadius());
        System.out.println("Area: " + circle2.getArea());
        System.out.println("\nCircle 3:");
        System.out.println("Radius: " + circle3.getRadius());
        System.out.println("Area: " + circle3.getArea());
    }
}
```

Output:



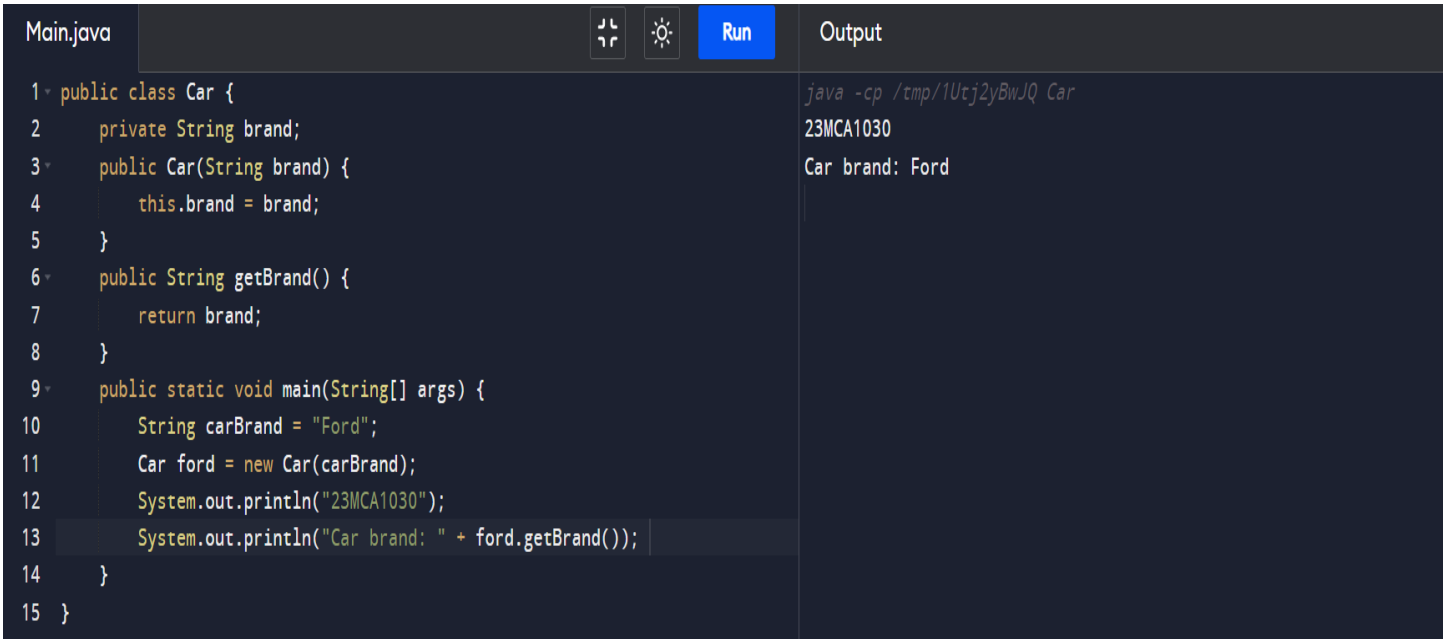
6.Write a constructor in the Car class given below that initializes the brand class field with the string "Ford". Call the getBrand() method in the main method of the Sample class and store the value of the brand in a variable, and print the value.

Code:

```
public class Car {
    private String brand;
    public Car(String brand) {
        this.brand = brand;
    }
    public String getBrand() {
        return brand;
    }
    public static void main(String[] args) {
        String carBrand = "Ford";
        Car ford = new Car(carBrand);
        System.out.println("23MCA1030");
        System.out.println("Car brand: " + ford.getBrand());
```

```
    }
}
```

Output:



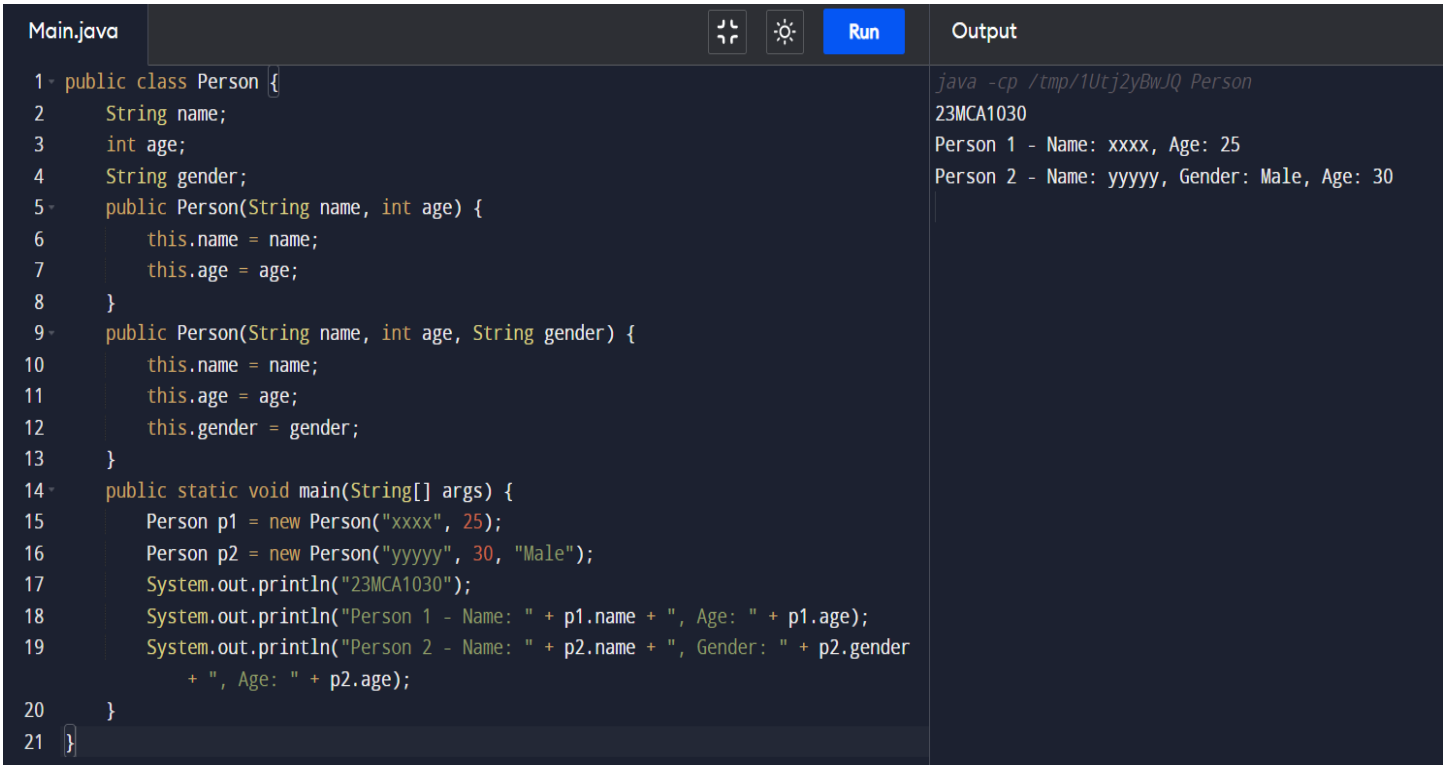
7. Write a Java program to create a class called "Person" with a name and age attribute. Create two instances of the "Person" class, set their attributes using the constructor, and print their name and age.

Code:

```
public class Person {
    String name;
    int age;
    String gender;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public Person(String name, int age, String gender) {
        this.name = name;
        this.age = age;
```

```
        this.gender = gender;
    }
    public static void main(String[] args) {
        Person p1 = new Person("xxxx", 25);
        Person p2 = new Person("yyyyy", 30, "Male");
        System.out.println("23MCA1030");
        System.out.println("Person 1 - Name: " + p1.name + ", Age: " + p1.age);
        System.out.println("Person 2 - Name: " + p2.name + ", Gender: " + p2.gender + ", Age: " + p2.age);
    }
}
```

Output:



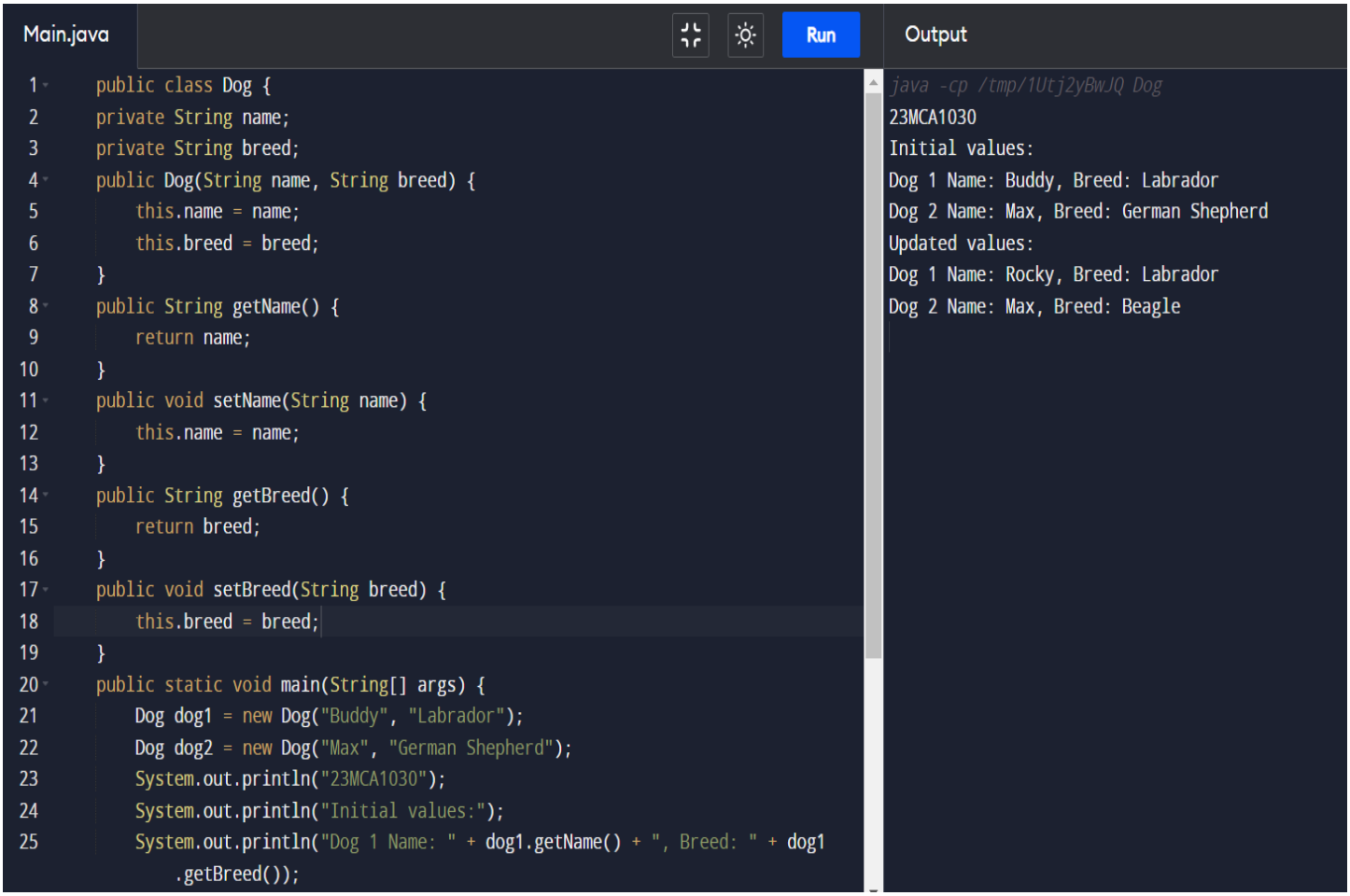
8. Write a Java program to create a class called "Dog" with a name and breed attribute. Create two instances of the "Dog" class, set their attributes using the constructor and modify the attributes using the setter methods and print the updated values

Code:

```
public class Dog {
private String name;
private String breed;
public Dog(String name, String breed) {
    this.name = name;
    this.breed = breed;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getBreed() {
    return breed;
}
public void setBreed(String breed) {
    this.breed = breed;
}
public static void main(String[] args) {
    Dog dog1 = new Dog("Buddy", "Labrador");
    Dog dog2 = new Dog("Max", "German Shepherd");
    System.out.println("23MCA1030");
```

```
        System.out.println("Initial values:");

        System.out.println("Dog 1 Name: " + dog1.getName() + ", Breed: " +
dog1.getBreed());

        System.out.println("Dog 2 Name: " + dog2.getName() + ", Breed: " +
dog2.getBreed());

        dog1.setName("Rocky");

        dog2.setBreed("Beagle");

        System.out.println("\nUpdated values:");

        System.out.println("Dog 1 Name: " + dog1.getName() + ", Breed: " +
dog1.getBreed());

        System.out.println("Dog 2 Name: " + dog2.getName() + ", Breed: " +
dog2.getBreed());

    }

}
```

Output:



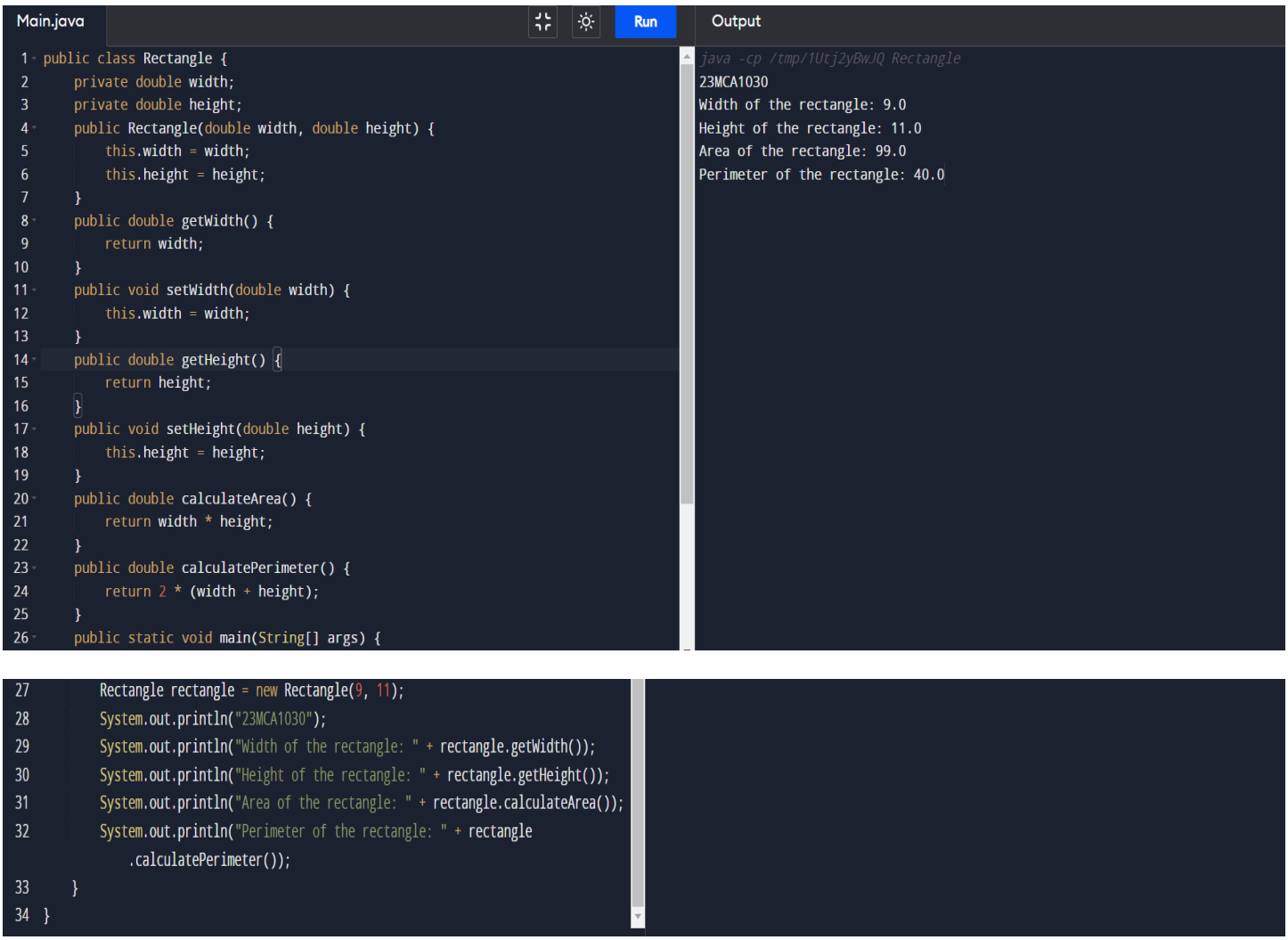
9. Write a Java program to create a class called "Rectangle" with width and height attributes. Calculate the area and perimeter of the rectangle.

Code:

```
public class Rectangle {
    private double width;
    private double height;
    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }
    public double getWidth() {
        return width;
    }
    public void setWidth(double width) {
        this.width = width;
    }
    public double getHeight() {
        return height;
    }
    public void setHeight(double height) {
        this.height = height;
    }
    public double calculateArea() {
        return width * height;
    }
    public double calculatePerimeter() {
        return 2 * (width + height);
```

```
    }
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(9, 11);
        System.out.println("23MCA1030");
        System.out.println("Width of the rectangle: " + rectangle.getWidth());
        System.out.println("Height of the rectangle: " + rectangle.getHeight());
        System.out.println("Area of the rectangle: " + rectangle.calculateArea());
        System.out.println("Perimeter of the rectangle: " +
rectangle.calculatePerimeter());
    }
}
```

Output:



10. Write a Java program to create a class called "Circle" with a radius attribute. You can access and modify this attribute. Calculate the area and circumference of the circle.

Code:

```
public class Circle {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    public double getRadius() {
        return radius;
    }
    public void setRadius(double radius) {
        this.radius = radius;
    }
    public double calculateArea() {
        return Math.PI * radius * radius;
    }
    public double calculateCircumference() {
        return 2 * Math.PI * radius;
    }
    public static void main(String[] args) {
        Circle circle = new Circle(16);
        System.out.println("23MCA1030");
        System.out.println("Radius of the circle: " + circle.getRadius());
        System.out.println("Area of the circle: " + circle.calculateArea());
```

```
        System.out.println("Circumference of the circle: " +
circle.calculateCircumference());
    }
}
```

Output: