...for a complete human being

Texas

International college

Java Assignment

Object Oriented Programming

Submitted By:

Niharika Shrestha

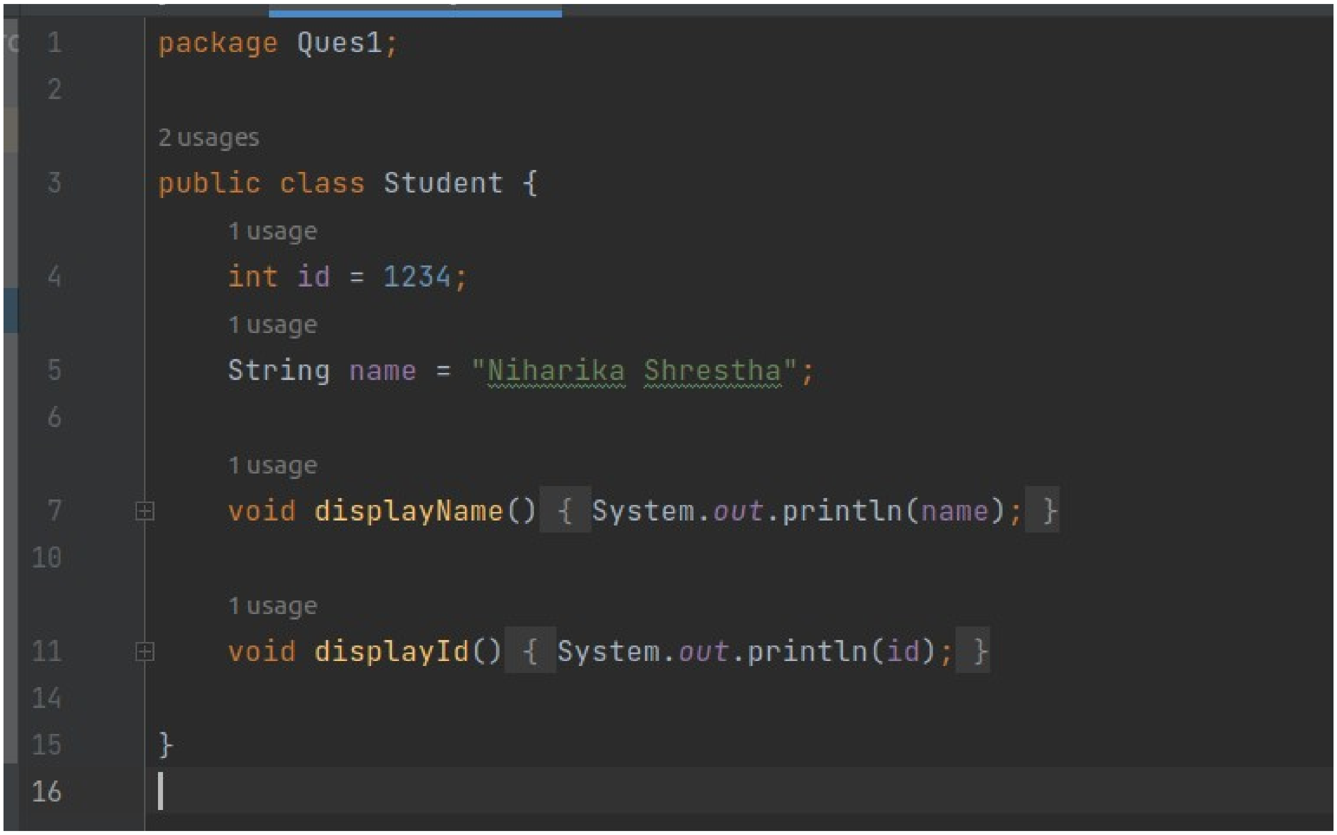
Submitted To:

Bishranta Bhattarai

Question no 1.
Write a program of Student and initializes it through reference variable that contains:
-Student id
-Student name
-Method to display Student name and id

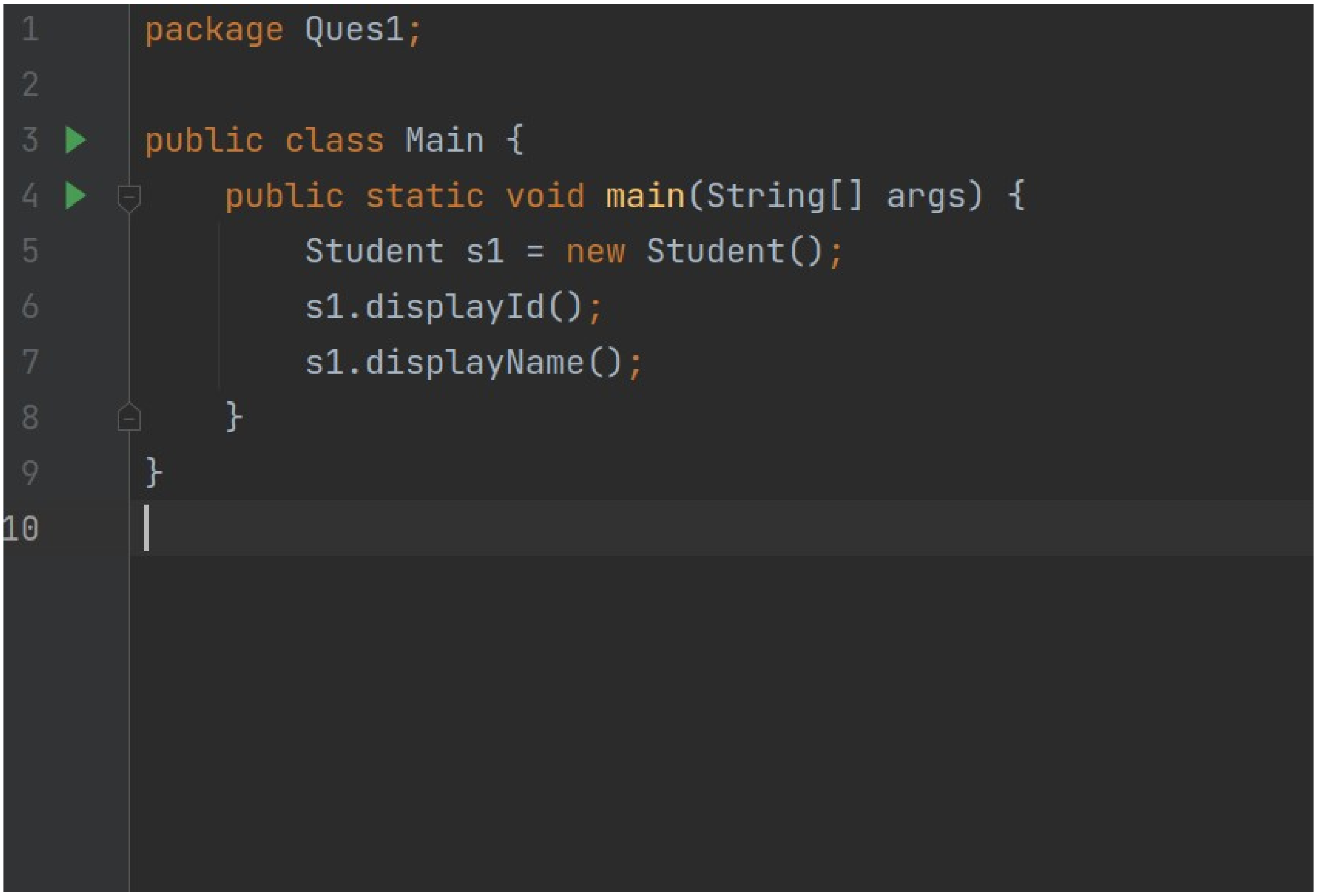
Ans:

<u>Student class</u>



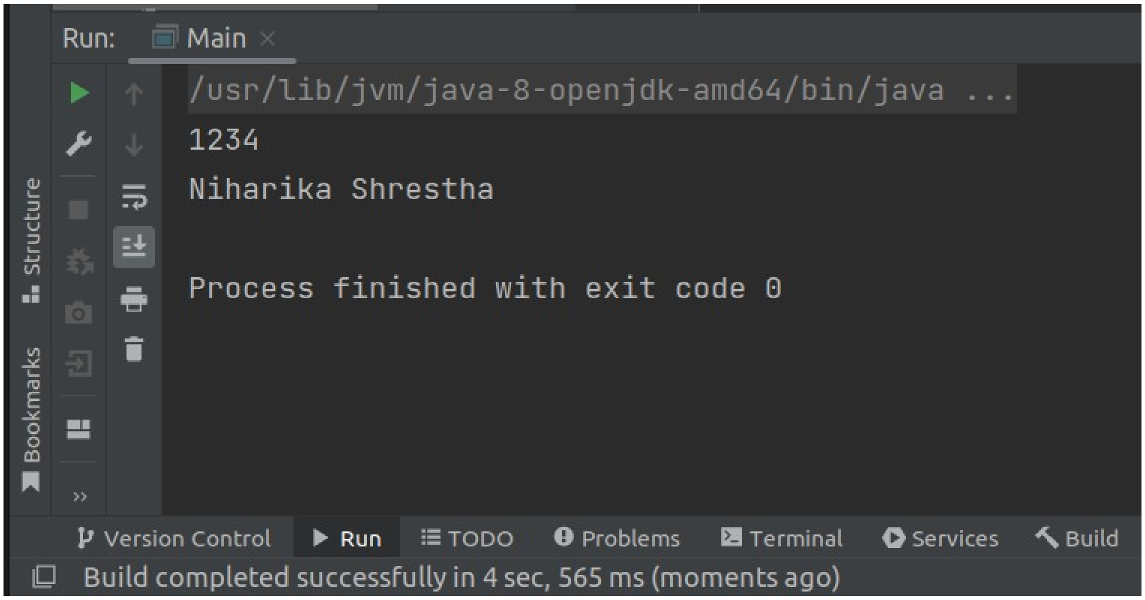
```java
package Ques1;

public class Student {
    int id = 1234;
    String name = "Niharika Shrestha";

    void displayName() { System.out.println(name); }

    void displayId() { System.out.println(id); }

}

```

Main file



```java
package Ques1;

public class Main {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.displayId();
        s1.displayName();
    }
}

```

Output:



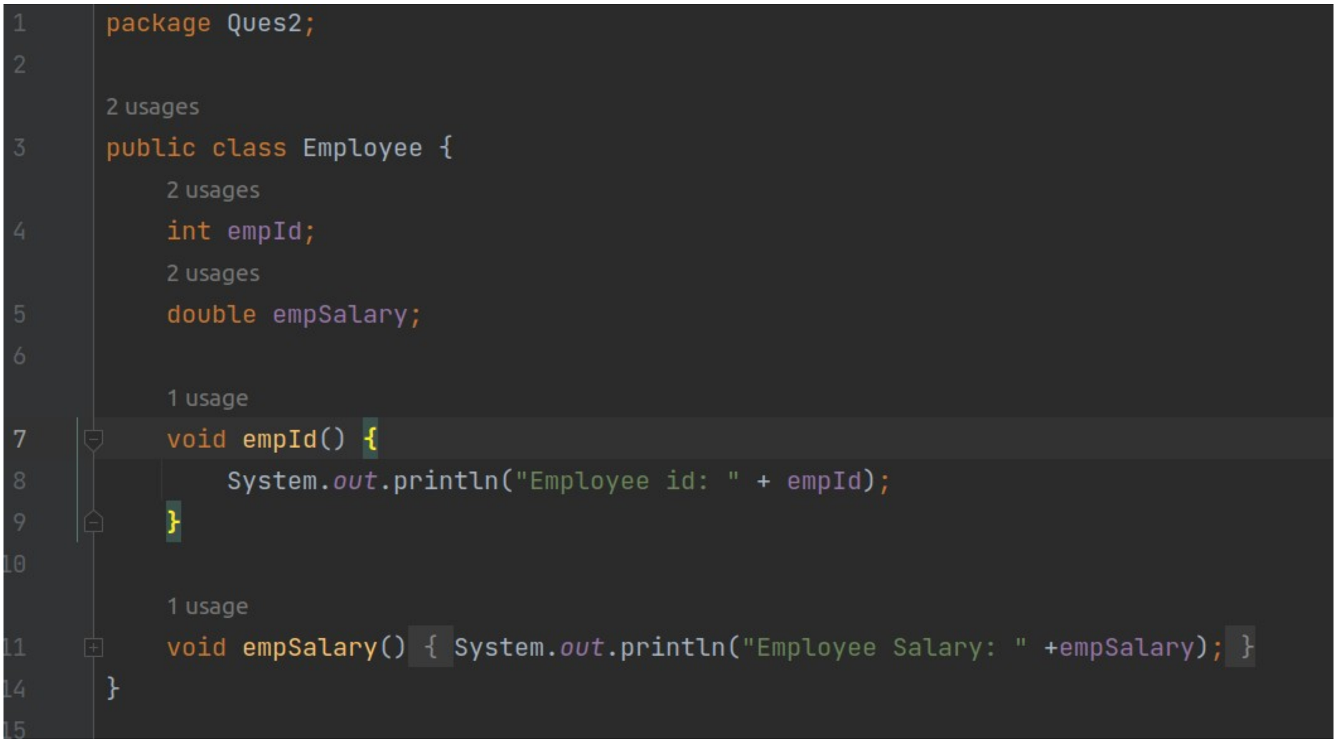
```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
1234
Niharika Shrestha

Process finished with exit code 0
```

Build completed successfully in 4 sec, 565 ms (moments ago)

Question no 2.
Write a program of Employee and initializes it through reference variable that contains:
-Employee Id
-Employee salary
- Method to display Employee Id and salary

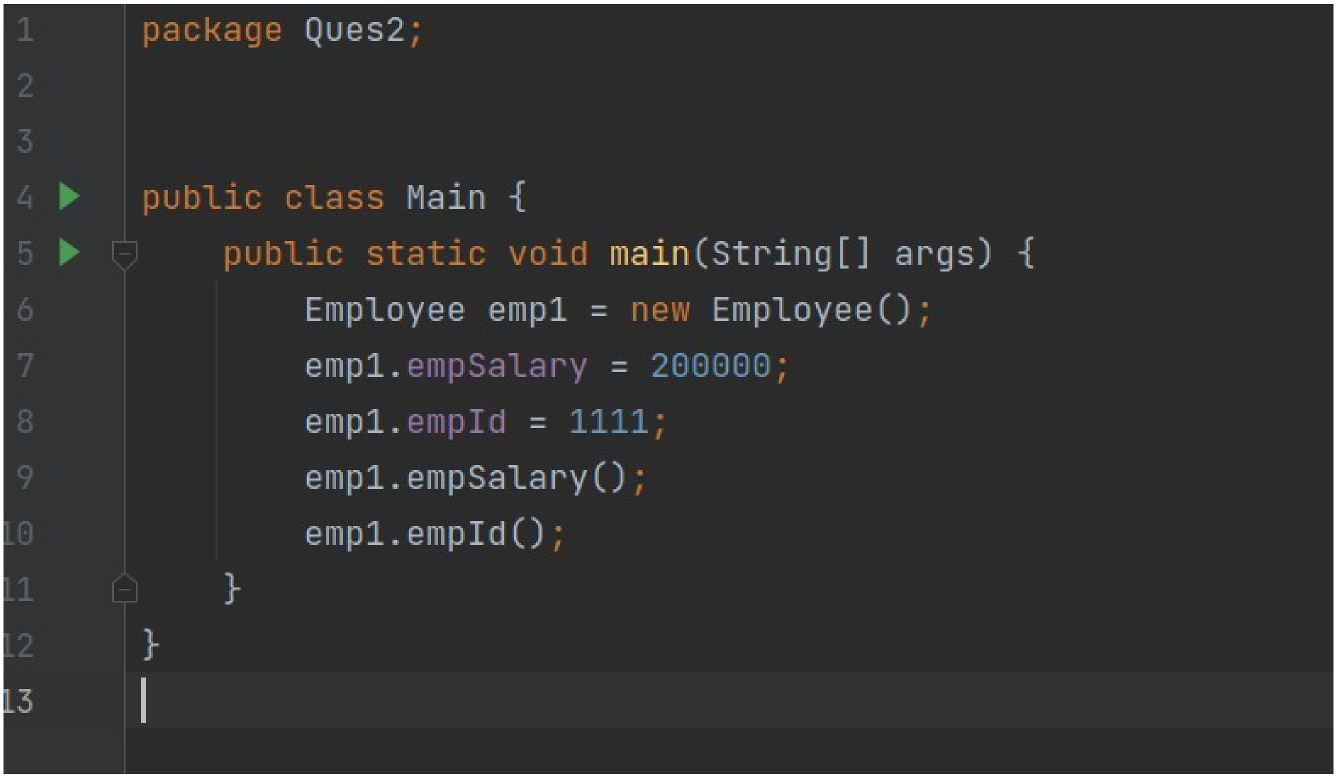
Ans:

Employee file



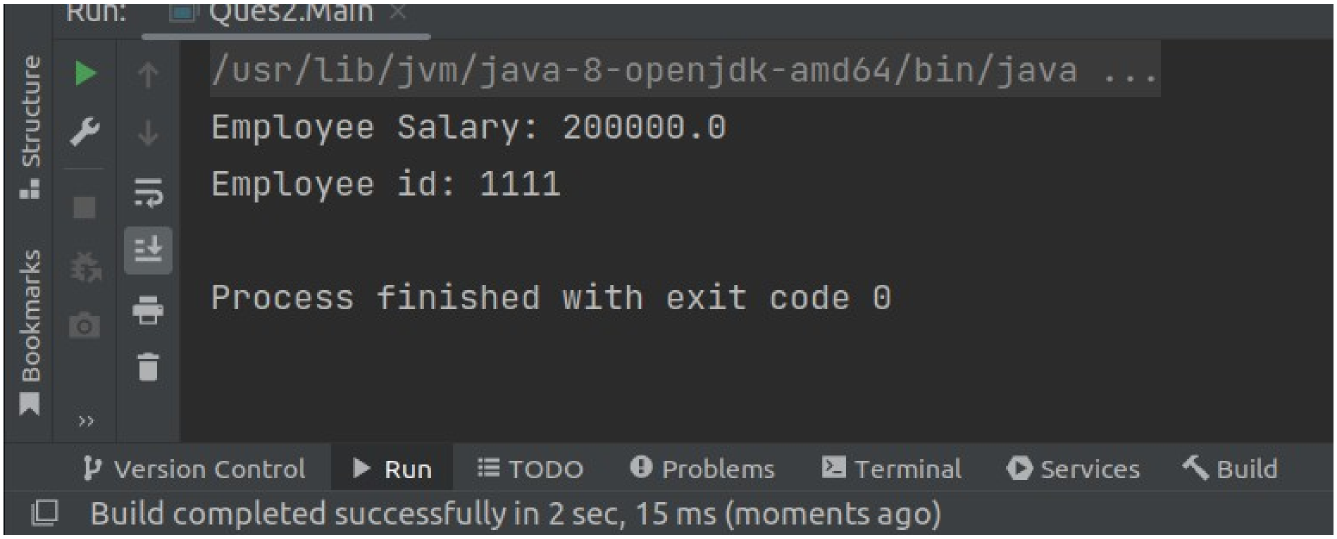
```
package Ques2;

2 usages
public class Employee {
    2 usages
    int empId;
    2 usages
    double empSalary;

    1 usage
    void empId() {
        System.out.println("Employee id: " + empId);
    }

    1 usage
    void empSalary() { System.out.println("Employee Salary: " +empSalary); }
}
```

Main file



```java
package Ques2;


public class Main {
    public static void main(String[] args) {
        Employee emp1 = new Employee();
        emp1.empSalary = 200000;
        emp1.empId = 1111;
        emp1.empSalary();
        emp1.empId();
    }
}

```

Output



```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Employee Salary: 200000.0
Employee id: 1111

Process finished with exit code 0
```

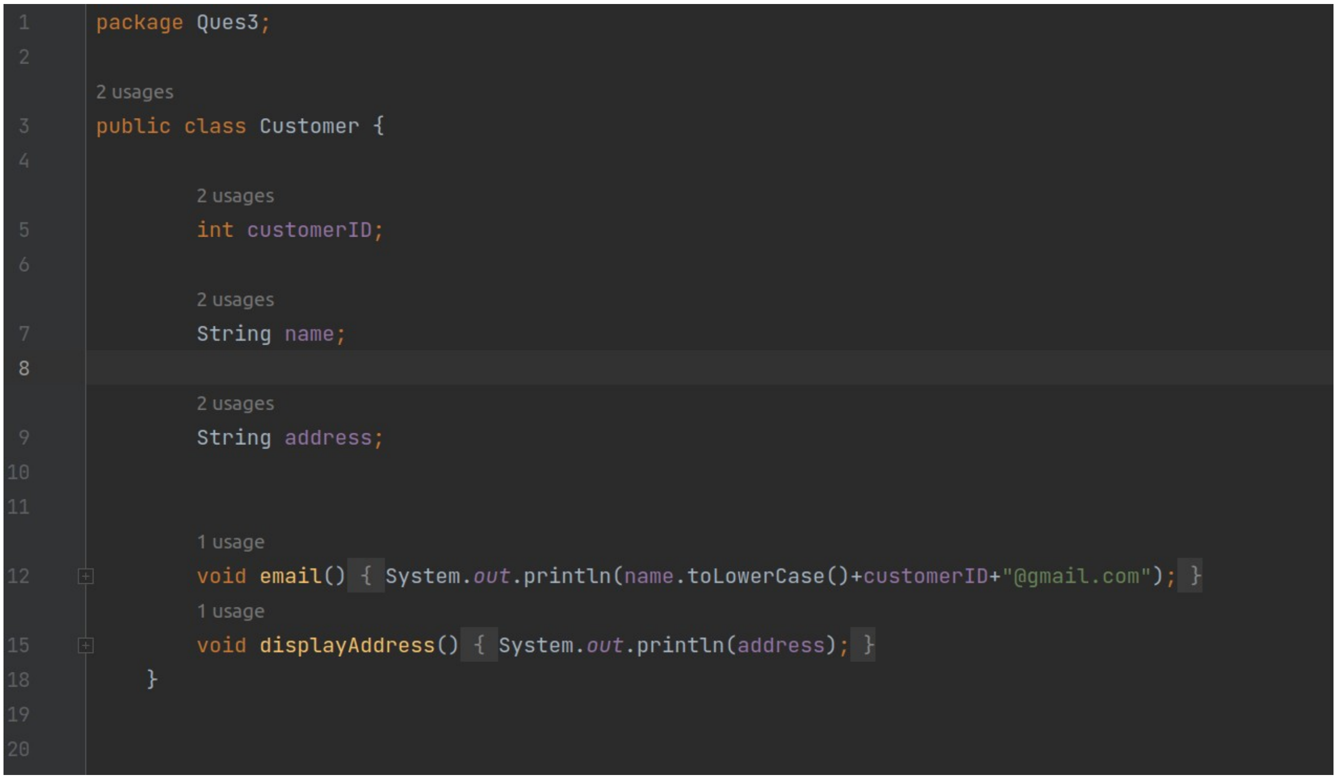
Build completed successfully in 2 sec, 15 ms (moments ago)

Ques 3: Write a program of Customer and initializes it through reference variable that contains:

- Customer Id
- Customer name
- Customer address
- Method to generate auto Customer email through name

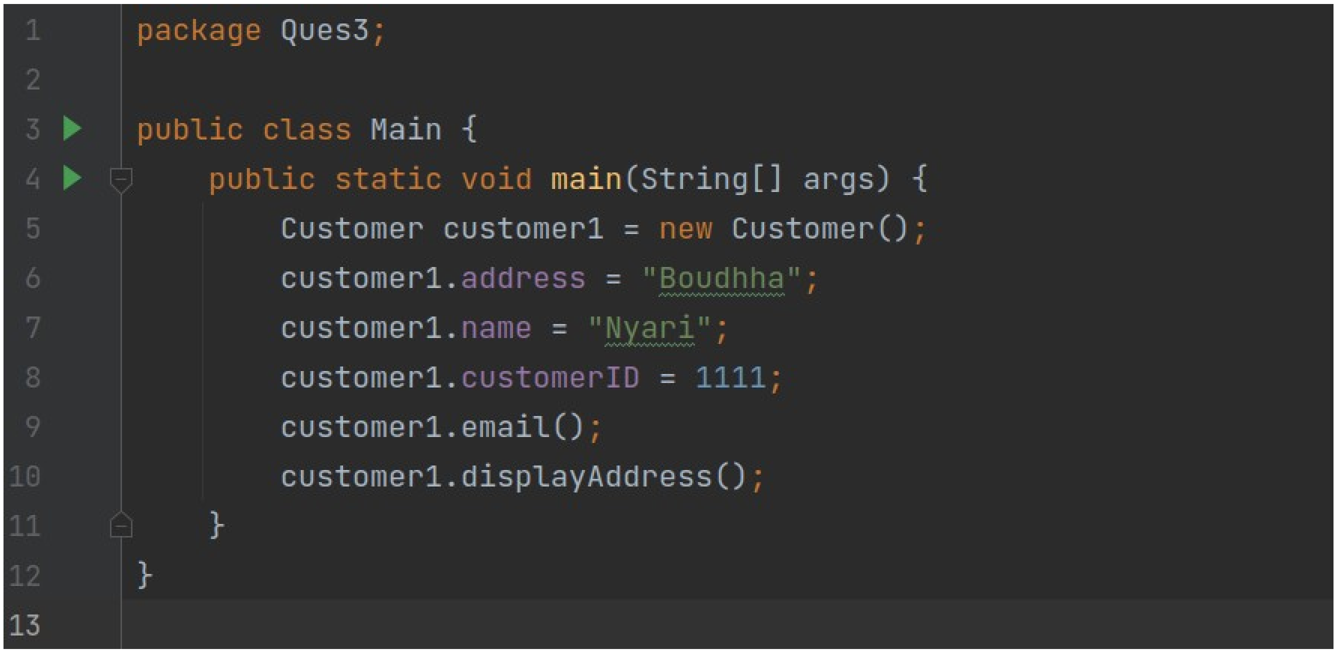
Ans :

<u>Customer File</u>



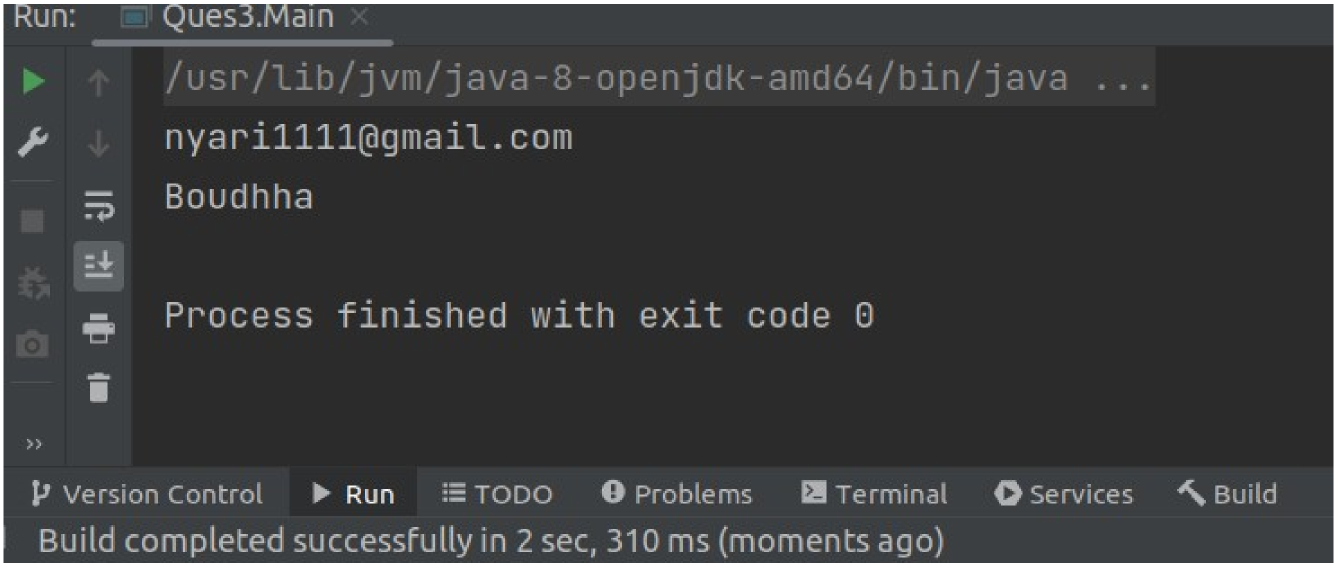
```
package Ques3;

2 usages
public class Customer {

        2 usages
        int customerID;

        2 usages
        String name;

        2 usages
        String address;


        1 usage
        void email() { System.out.println(name.toLowerCase()+customerID+"@gmail.com"); }
        1 usage
        void displayAddress() { System.out.println(address); }
    }


```

Main file



```java
package Ques3;

public class Main {
    public static void main(String[] args) {
        Customer customer1 = new Customer();
        customer1.address = "Boudhha";
        customer1.name = "Nyari";
        customer1.customerID = 1111;
        customer1.email();
        customer1.displayAddress();
    }
}
```

Output



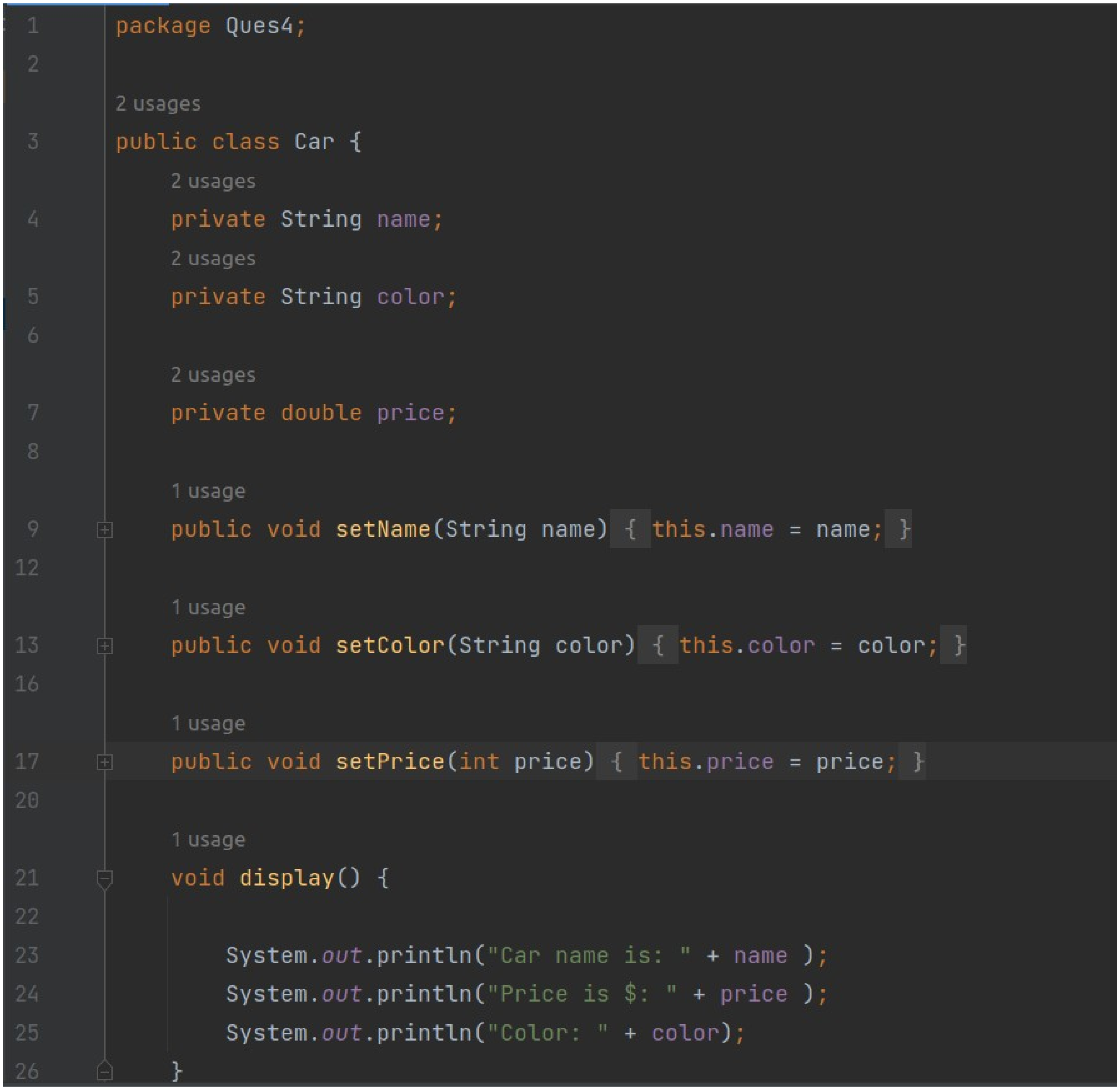
```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
nyari1111@gmail.com
Boudhha

Process finished with exit code 0
```

Build completed successfully in 2 sec, 310 ms (moments ago)

Q no 4: Write a program of Car and initializes it through Method that contains:
- Car name
- Car color
- Car price
- Method to display Car name and price

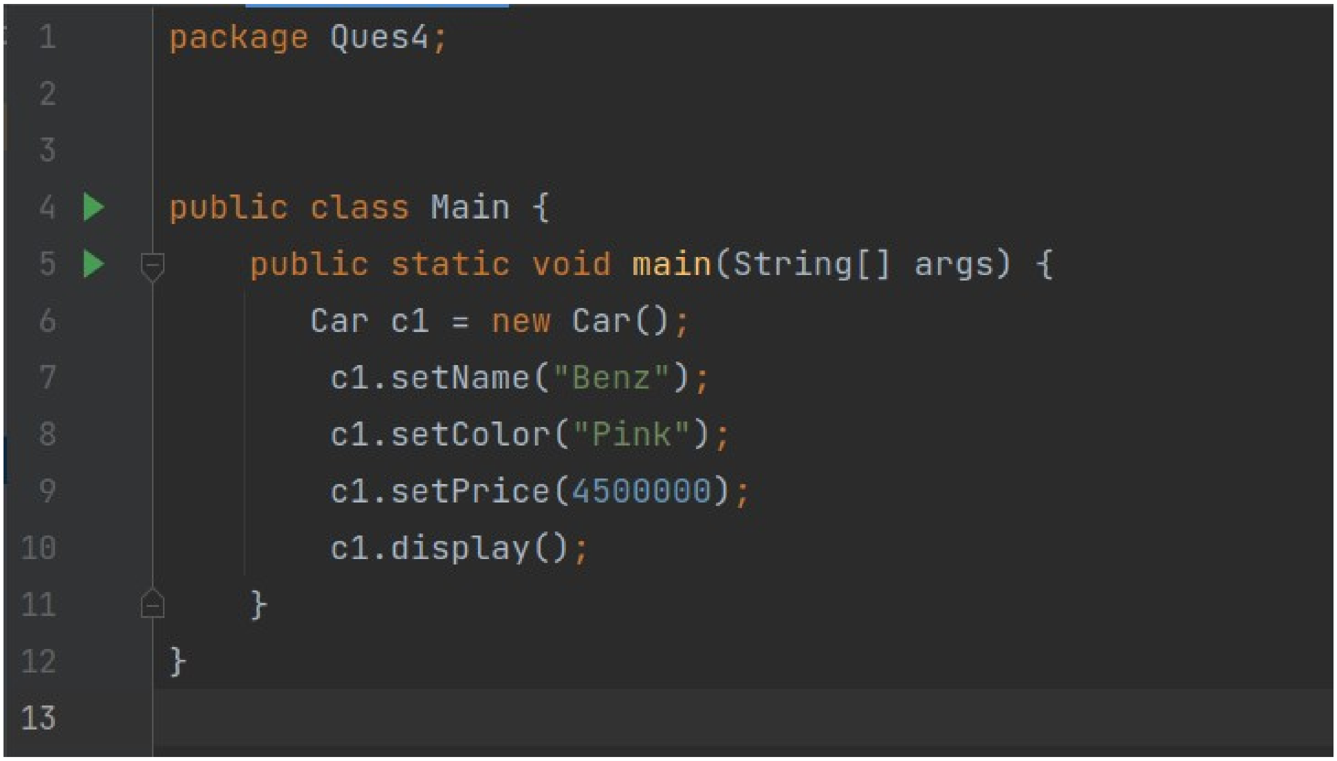
Ans:

<u>Car File</u>



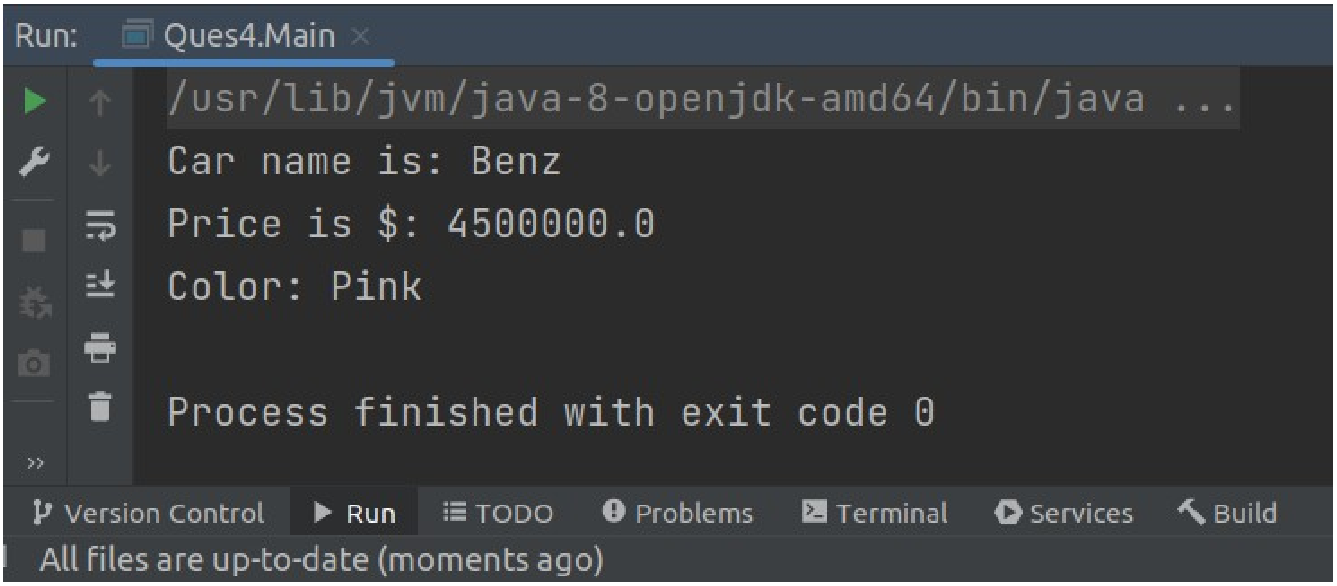
```
package Ques4;

2 usages
public class Car {
    2 usages
    private String name;
    2 usages
    private String color;

    2 usages
    private double price;

    1 usage
    public void setName(String name) { this.name = name; }

    1 usage
    public void setColor(String color) { this.color = color; }

    1 usage
    public void setPrice(int price) { this.price = price; }

    1 usage
    void display() {

        System.out.println("Car name is: " + name );
        System.out.println("Price is $: " + price );
        System.out.println("Color: " + color);
    }
```

Main file



```java
package Ques4;


public class Main {
    public static void main(String[] args) {
        Car c1 = new Car();
         c1.setName("Benz");
         c1.setColor("Pink");
         c1.setPrice(4500000);
         c1.display();
    }
}

```

Output:



```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Car name is: Benz
Price is $: 4500000.0
Color: Pink

Process finished with exit code 0
```

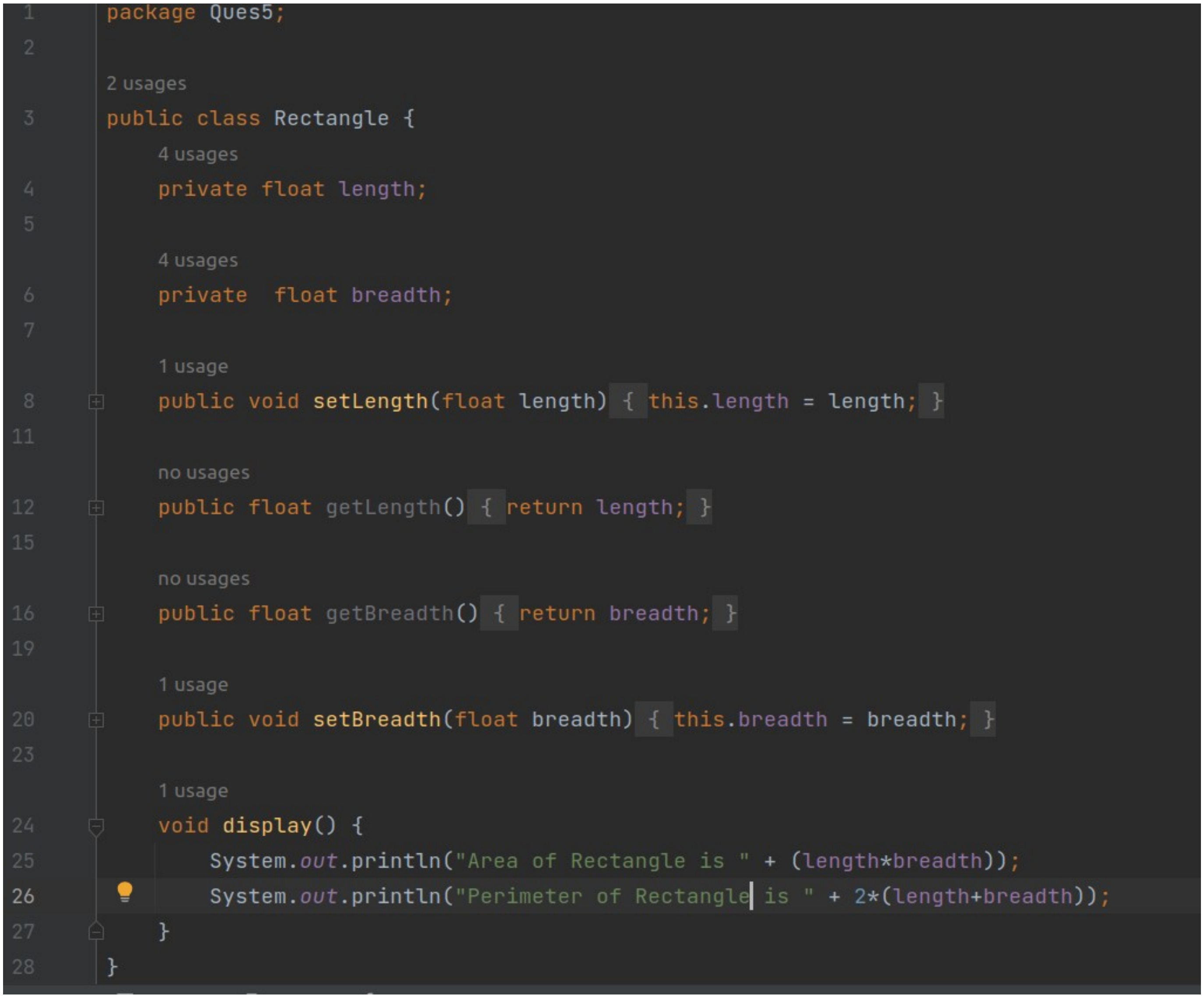
All files are up-to-date (moments ago)

Ques no 5: Write a program of Rectangle and initializes it through Method that contains:

- Rectangle length
- Rectangle breadth
- Method to display area and perimeter

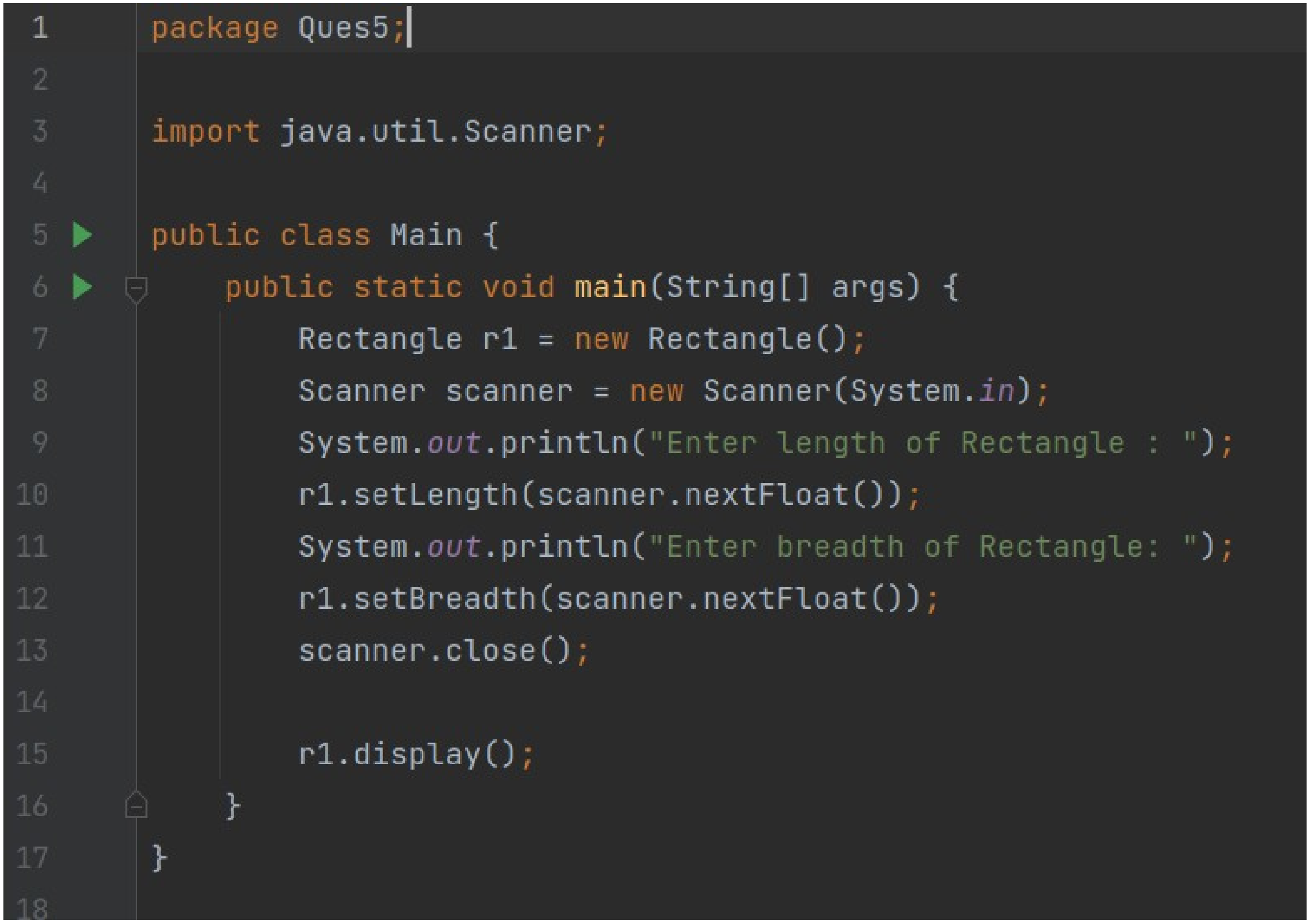
Ans:

<u>Rectangle Class file</u>



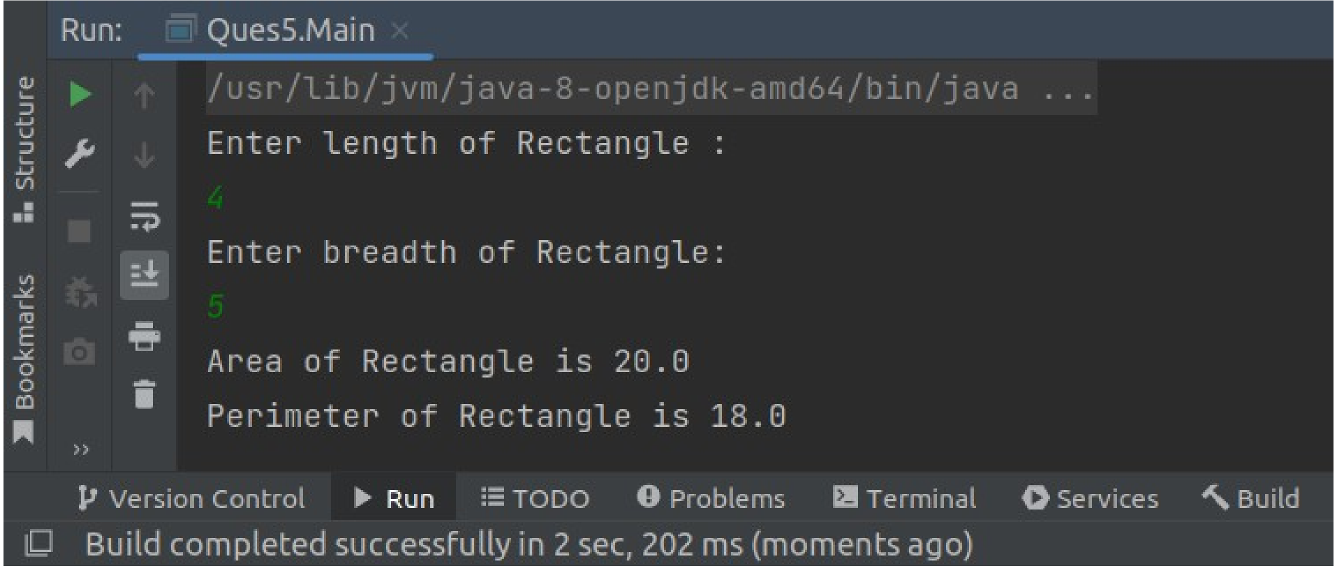
```
package Ques5;

2 usages
public class Rectangle {
    4 usages
    private float length;

    4 usages
    private  float breadth;

    1 usage
    public void setLength(float length) { this.length = length; }

    no usages
    public float getLength() { return length; }

    no usages
    public float getBreadth() { return breadth; }

    1 usage
    public void setBreadth(float breadth) { this.breadth = breadth; }

    1 usage
    void display() {
        System.out.println("Area of Rectangle is " + (length*breadth));
        System.out.println("Perimeter of Rectangle is " + 2*(length+breadth));
    }
}
```

Main File



```java
package Ques5;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Rectangle r1 = new Rectangle();
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter length of Rectangle : ");
        r1.setLength(scanner.nextFloat());
        System.out.println("Enter breadth of Rectangle: ");
        r1.setBreadth(scanner.nextFloat());
        scanner.close();

        r1.display();
    }
}
```

Output



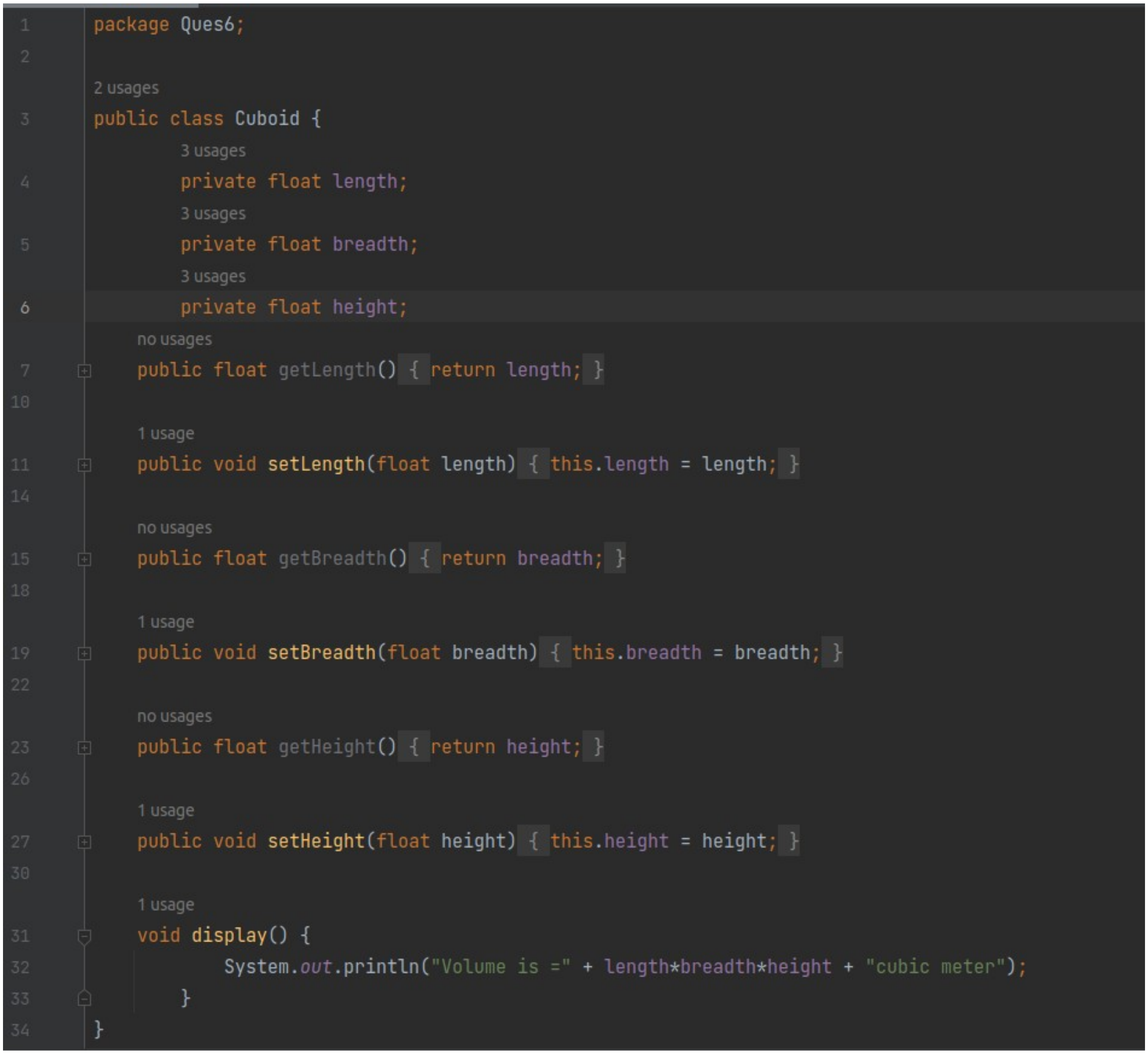
```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Enter length of Rectangle :
4
Enter breadth of Rectangle:
5
Area of Rectangle is 20.0
Perimeter of Rectangle is 18.0
```

Build completed successfully in 2 sec, 202 ms (moments ago)

Q no 6: Write a program of Cuboid and initializes it through Method that contains:
- Cuboid length
- Cuboid breadth
- Cuboid height
- Method to display volume of Cuboid

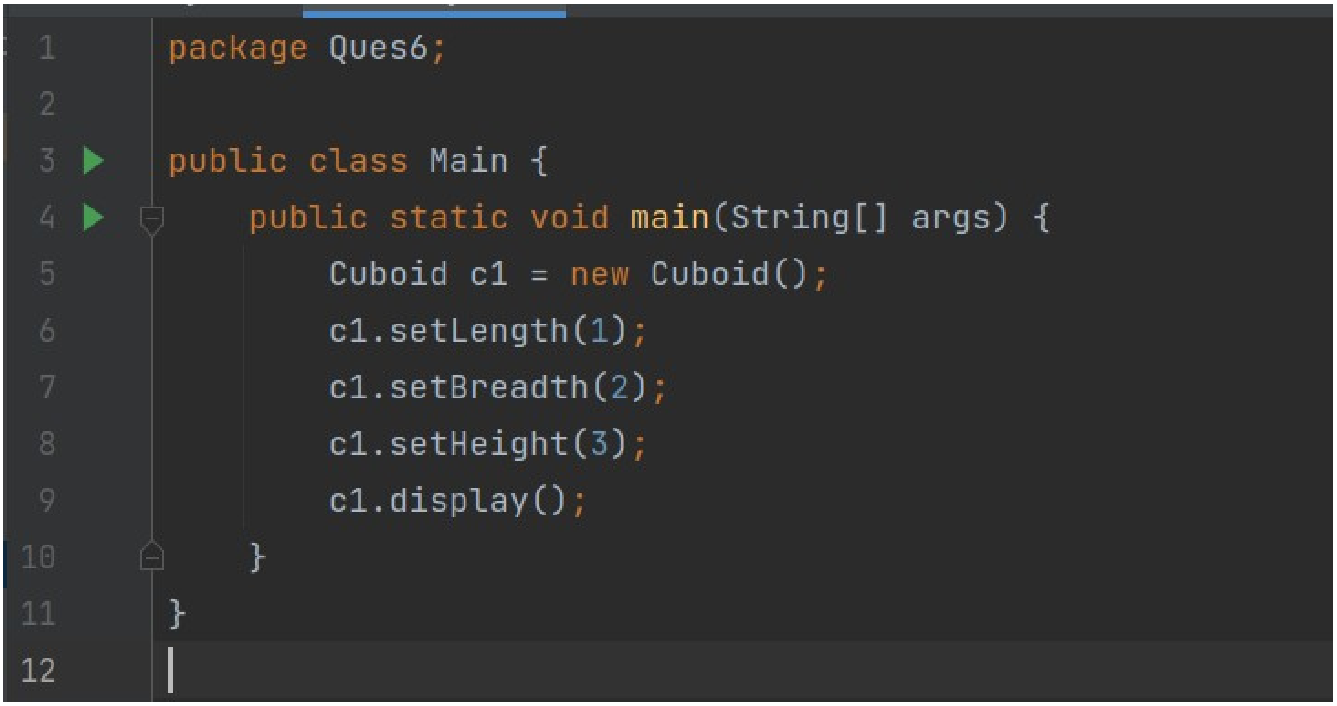
Ans:

<u>Cuboid Class File</u>



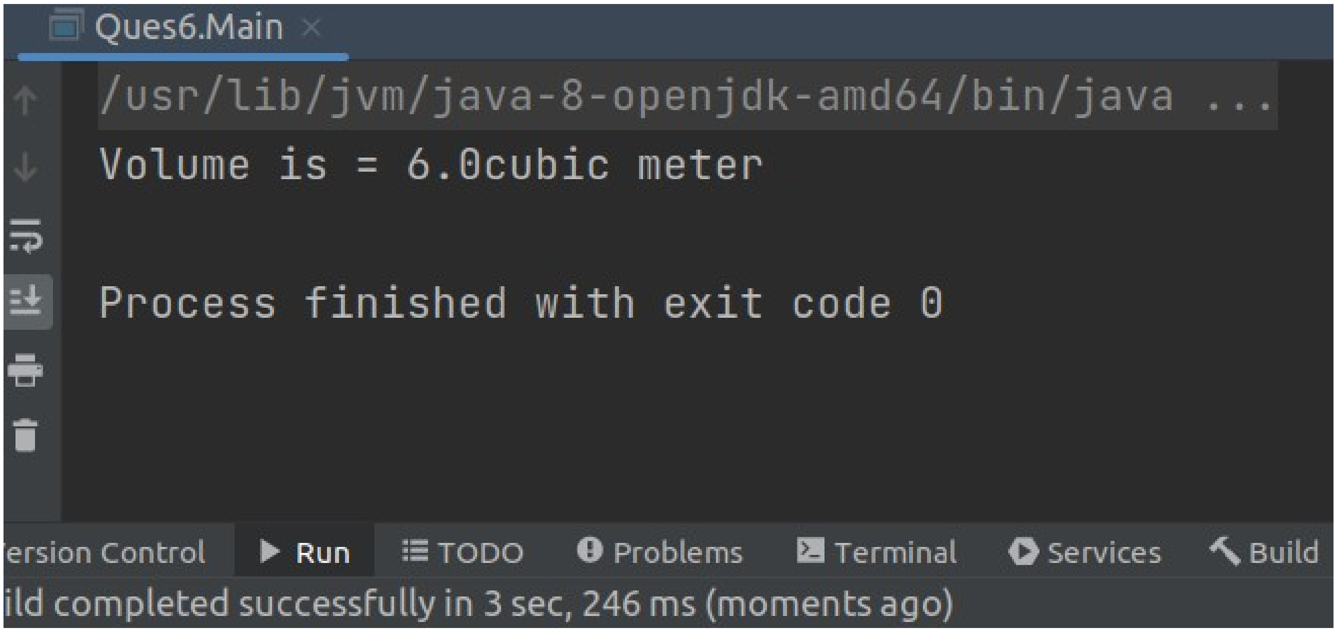
```
package Ques6;

2 usages
public class Cuboid {
        3 usages
        private float length;
        3 usages
        private float breadth;
        3 usages
        private float height;
    no usages
    public float getLength() { return length; }

    1 usage
    public void setLength(float length) { this.length = length; }

    no usages
    public float getBreadth() { return breadth; }

    1 usage
    public void setBreadth(float breadth) { this.breadth = breadth; }

    no usages
    public float getHeight() { return height; }

    1 usage
    public void setHeight(float height) { this.height = height; }

    1 usage
    void display() {
            System.out.println("Volume is =" + length*breadth*height + "cubic meter");
        }
}
```

Main file



```java
package Ques6;

public class Main {
    public static void main(String[] args) {
        Cuboid c1 = new Cuboid();
        c1.setLength(1);
        c1.setBreadth(2);
        c1.setHeight(3);
        c1.display();
    }
}

```

Output:



```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Volume is = 6.0cubic meter

Process finished with exit code 0
```

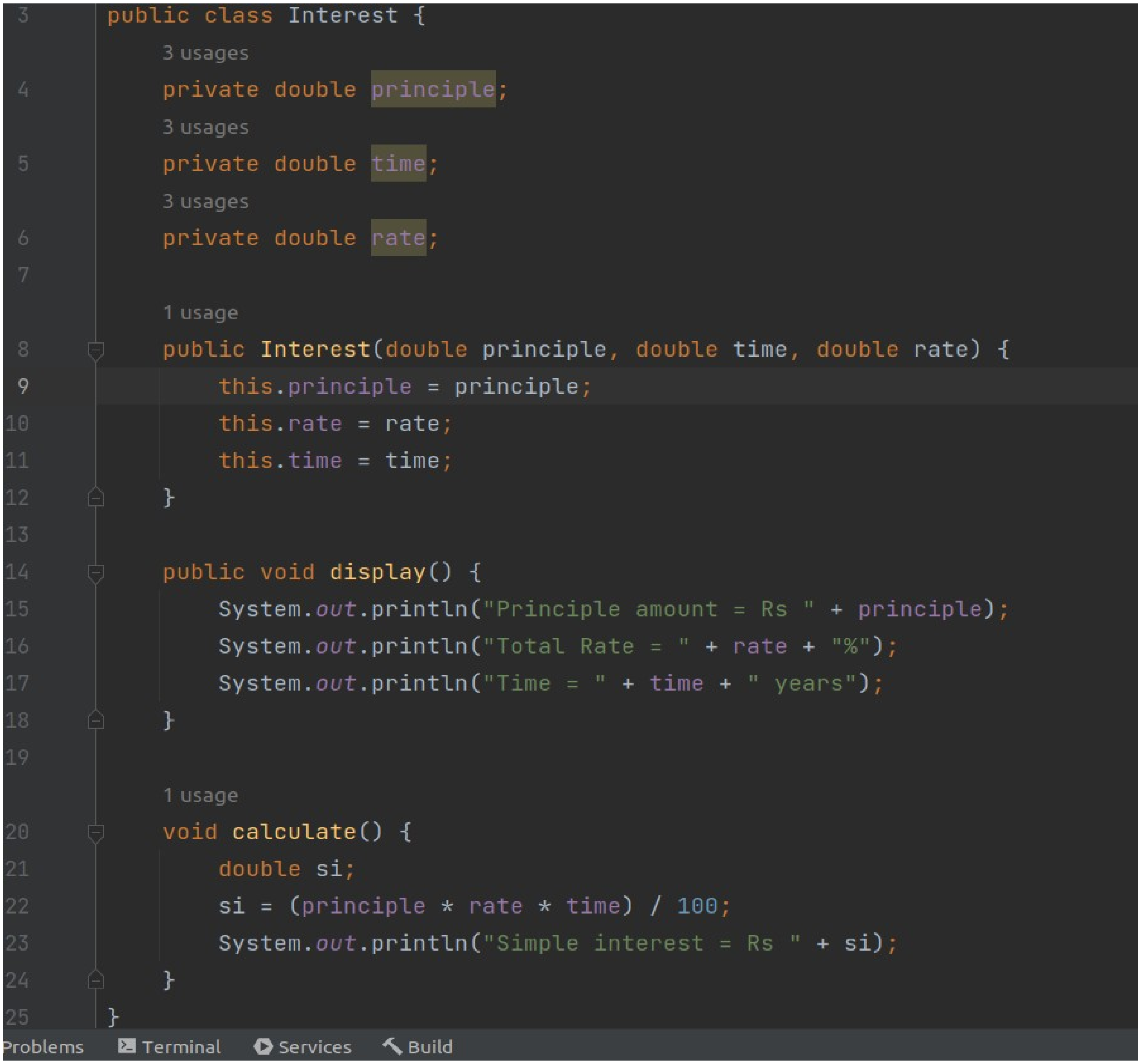
ild completed successfully in 3 sec, 246 ms (moments ago)

Ques no 7: Write a program of Interest and initializes it through Constructor that contains:

- Interest principle
- Interest time
- Interest rate
- Method to display Interest principle, time and rate
- Method to calculate and display simple interest (SI)

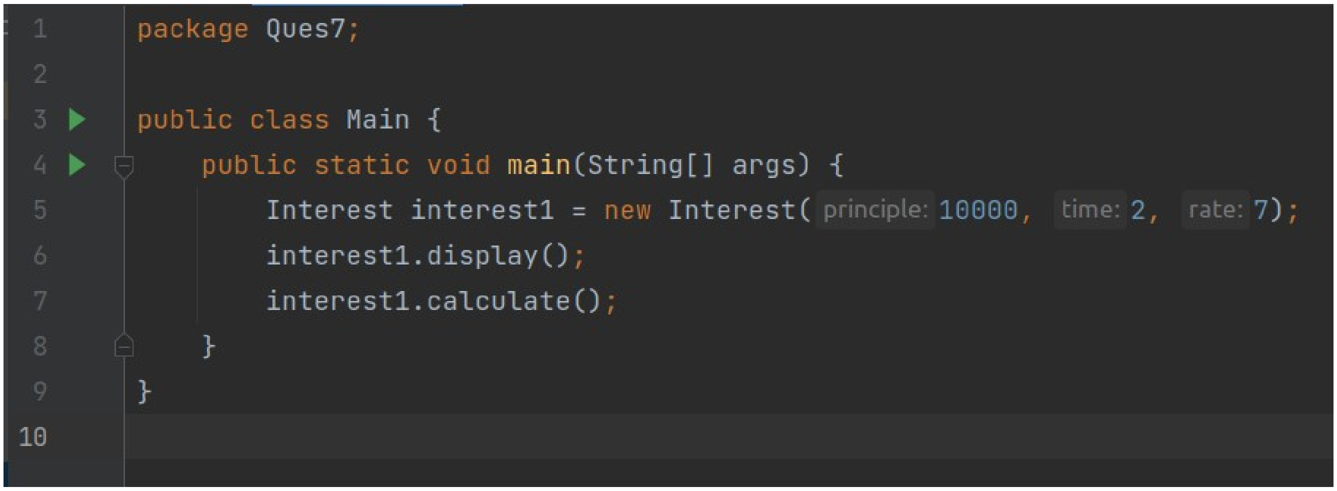
Ans:

<u>Interest Class File</u>



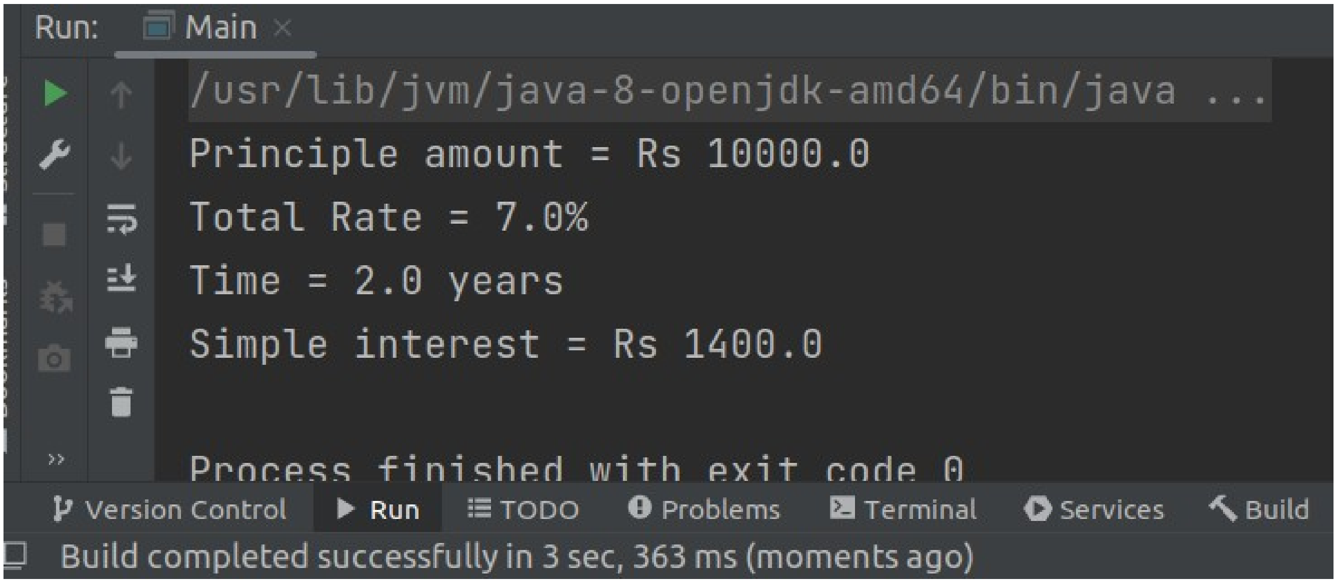
```
public class Interest {
    3 usages
    private double principle;
    3 usages
    private double time;
    3 usages
    private double rate;

    1 usage
    public Interest(double principle, double time, double rate) {
        this.principle = principle;
        this.rate = rate;
        this.time = time;
    }

    public void display() {
        System.out.println("Principle amount = Rs " + principle);
        System.out.println("Total Rate = " + rate + "%");
        System.out.println("Time = " + time + " years");
    }

    1 usage
    void calculate() {
        double si;
        si = (principle * rate * time) / 100;
        System.out.println("Simple interest = Rs " + si);
    }
}
```

Main file



```java
package Ques7;

public class Main {
    public static void main(String[] args) {
        Interest interest1 = new Interest( principle: 10000,  time: 2,  rate: 7);
        interest1.display();
        interest1.calculate();
    }
}

```

Output:



```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Principle amount = Rs 10000.0
Total Rate = 7.0%
Time = 2.0 years
Simple interest = Rs 1400.0

Process finished with exit code 0
```

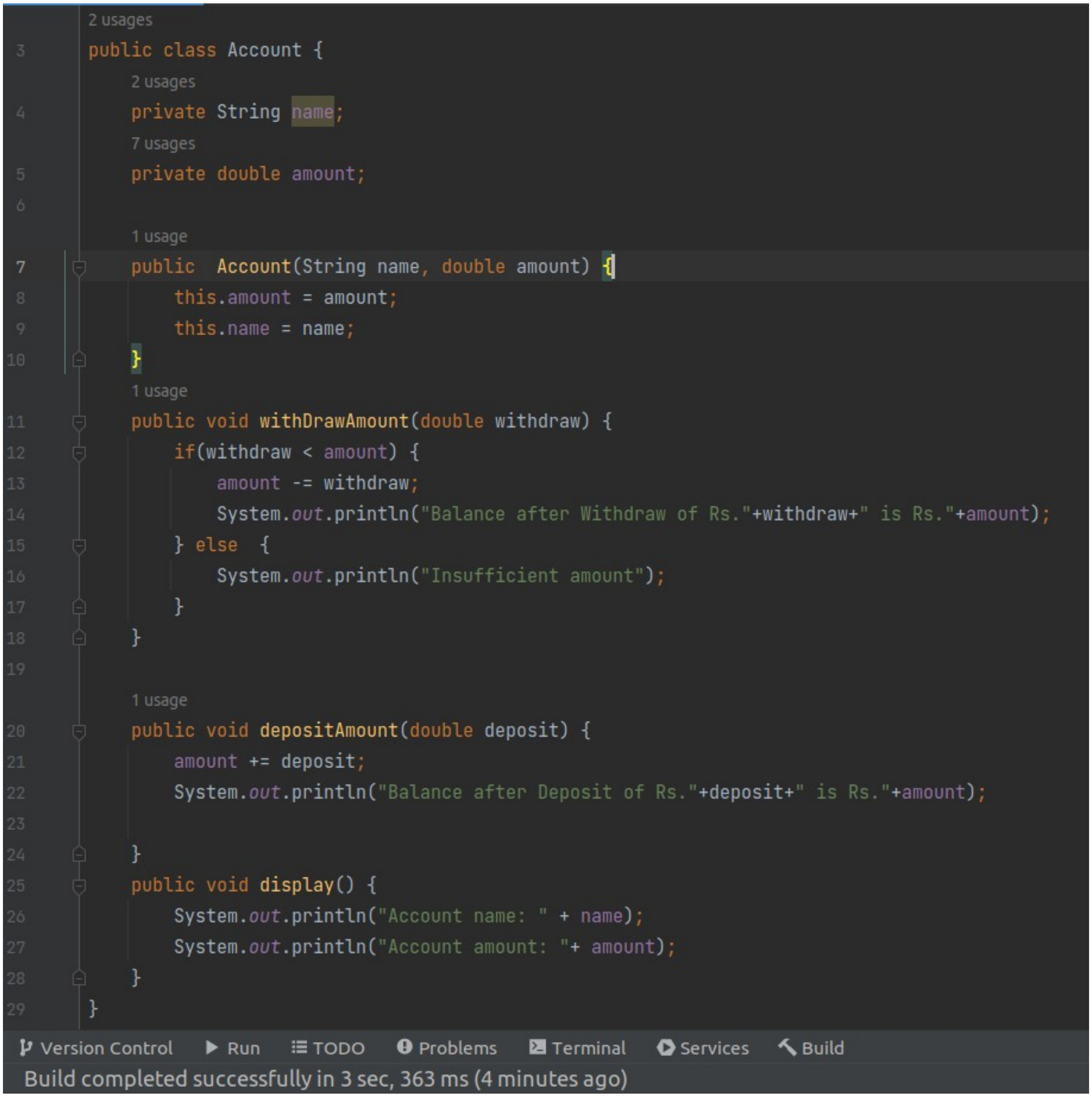
Build completed successfully in 3 sec, 363 ms (moments ago)

Ques no 8: Write a program of Account and initializes it through Constructor that contains:

- Account name
- Account amount
- Method to withdraw amount
- Method to deposit amount
- Method to display Account name and amount

Ans:

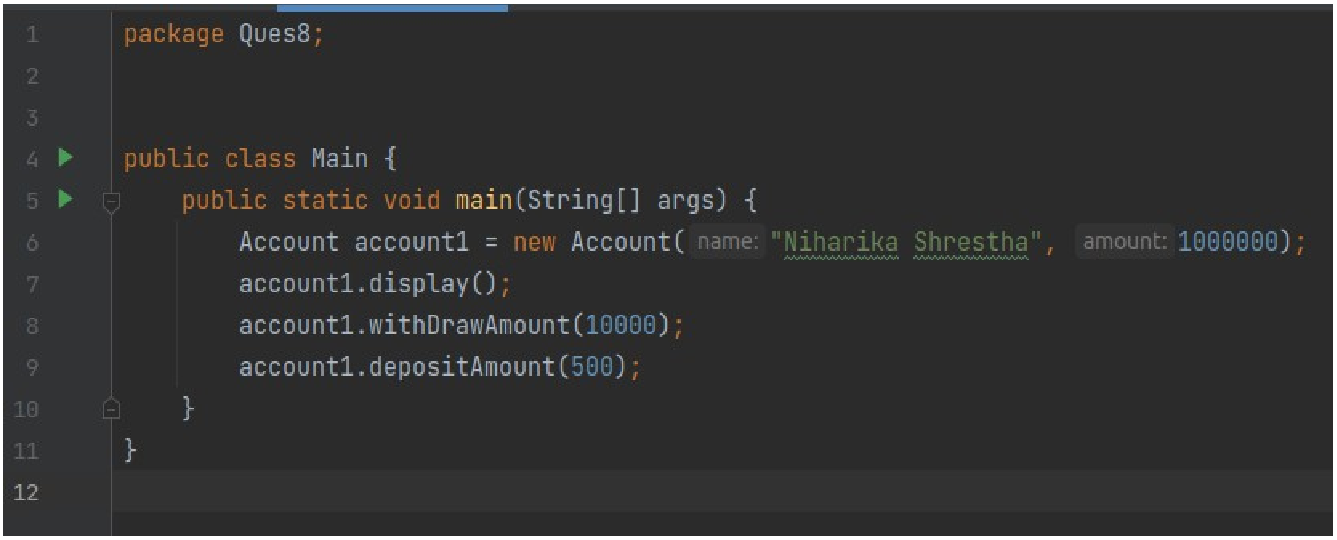
Account Class File



```java
public class Account {
    private String name;
    private double amount;

    public  Account(String name, double amount) {
        this.amount = amount;
        this.name = name;
    }
    public void withDrawAmount(double withdraw) {
        if(withdraw < amount) {
            amount -= withdraw;
            System.out.println("Balance after Withdraw of Rs."+withdraw+" is Rs."+amount);
        } else  {
            System.out.println("Insufficient amount");
        }
    }

    public void depositAmount(double deposit) {
        amount += deposit;
        System.out.println("Balance after Deposit of Rs."+deposit+" is Rs."+amount);

    }
    public void display() {
        System.out.println("Account name: " + name);
        System.out.println("Account amount: "+ amount);
    }
}
```

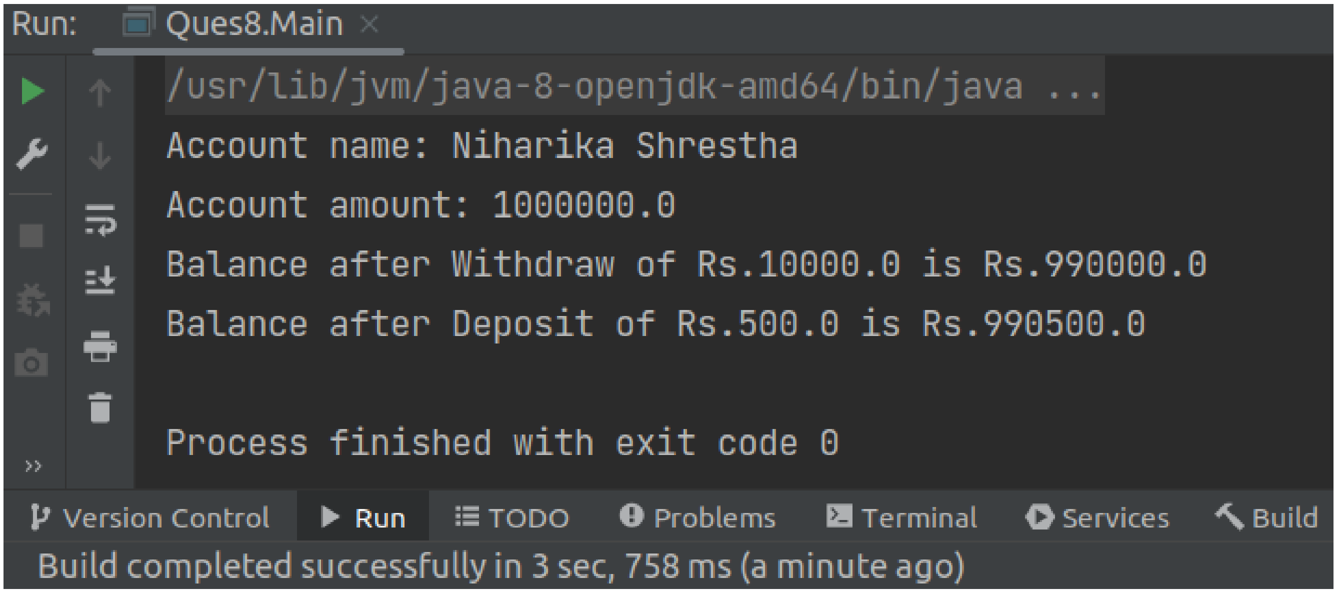
Build completed successfully in 3 sec, 363 ms (4 minutes ago)

Main File



```
package Ques8;


public class Main {
    public static void main(String[] args) {
        Account account1 = new Account( name: "Niharika Shrestha",  amount: 1000000);
        account1.display();
        account1.withDrawAmount(10000);
        account1.depositAmount(500);
    }
}

```

Output:



```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Account name: Niharika Shrestha
Account amount: 1000000.0
Balance after Withdraw of Rs.10000.0 is Rs.990000.0
Balance after Deposit of Rs.500.0 is Rs.990500.0

Process finished with exit code 0
```

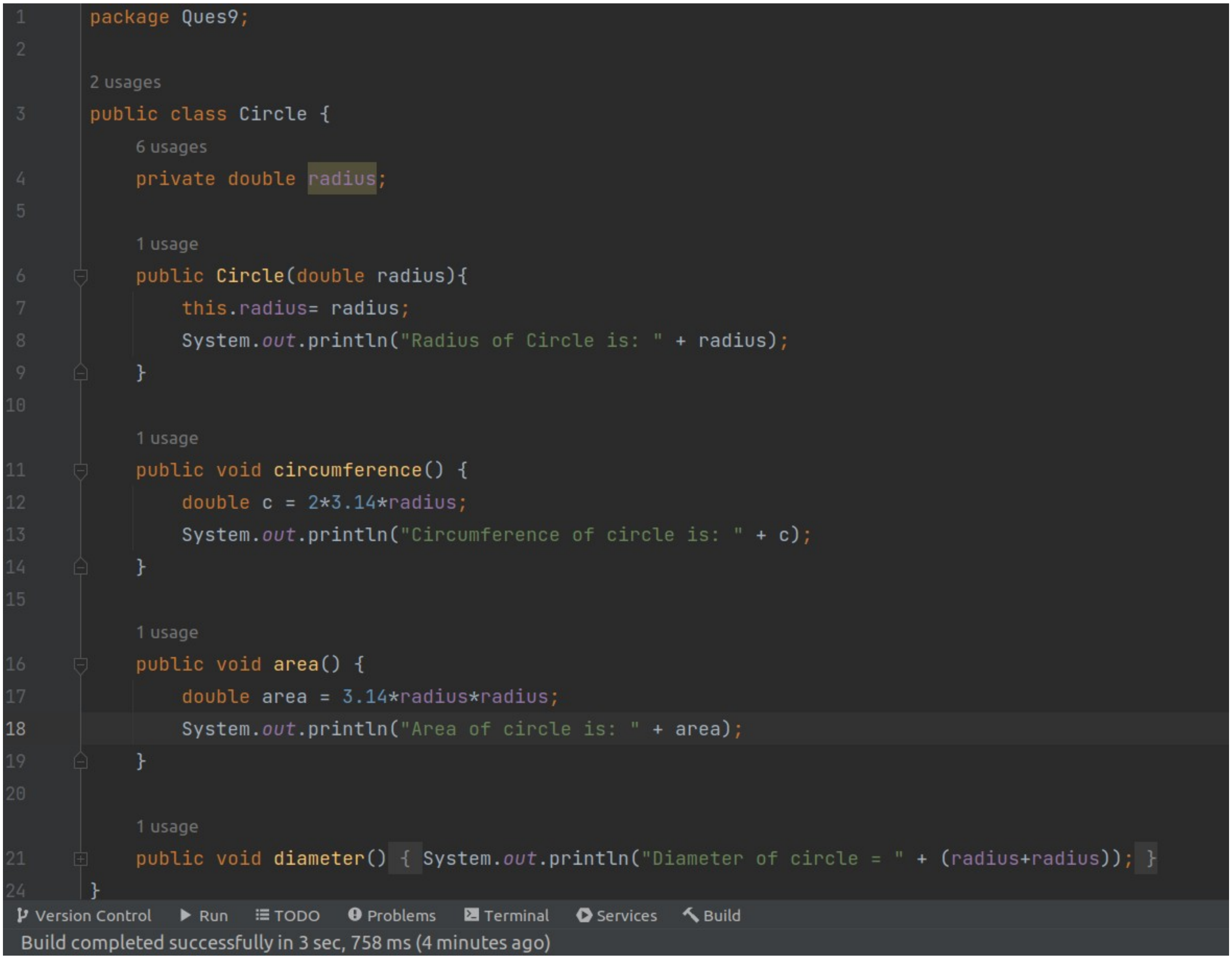
Build completed successfully in 3 sec, 758 ms (a minute ago)

Ques no 9: Write a program of Circle and initializes it through Constructor that contains:

- Circle radius
- Constructor to display Circle radius
- Method to calculate Circle diameter, perimeter and area.

Ans:

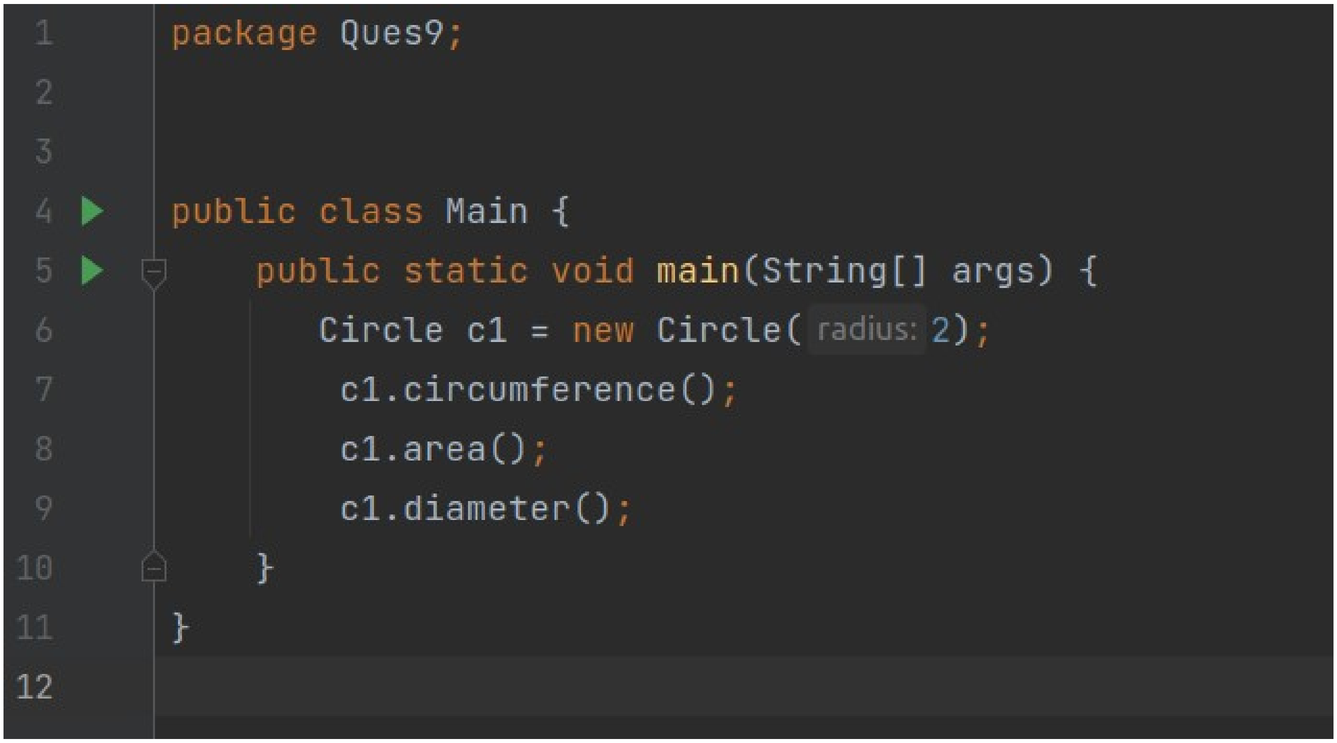
Circle Class File



```java
package Ques9;

public class Circle {
    private double radius;

    public Circle(double radius){
        this.radius= radius;
        System.out.println("Radius of Circle is: " + radius);
    }

    public void circumference() {
        double c = 2*3.14*radius;
        System.out.println("Circumference of circle is: " + c);
    }

    public void area() {
        double area = 3.14*radius*radius;
        System.out.println("Area of circle is: " + area);
    }

    public void diameter() { System.out.println("Diameter of circle = " + (radius+radius)); }
}
```

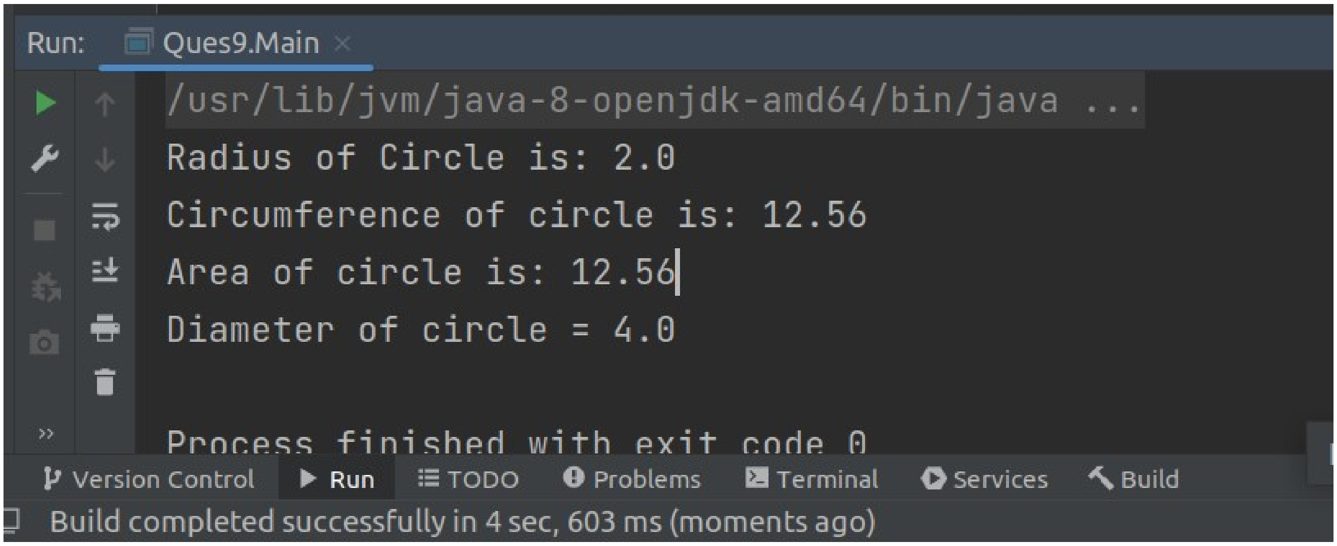
Build completed successfully in 3 sec, 758 ms (4 minutes ago)

Main File



```
package Ques9;


public class Main {
    public static void main(String[] args) {
        Circle c1 = new Circle( radius: 2);
         c1.circumference();
         c1.area();
         c1.diameter();
    }
}

```

Output

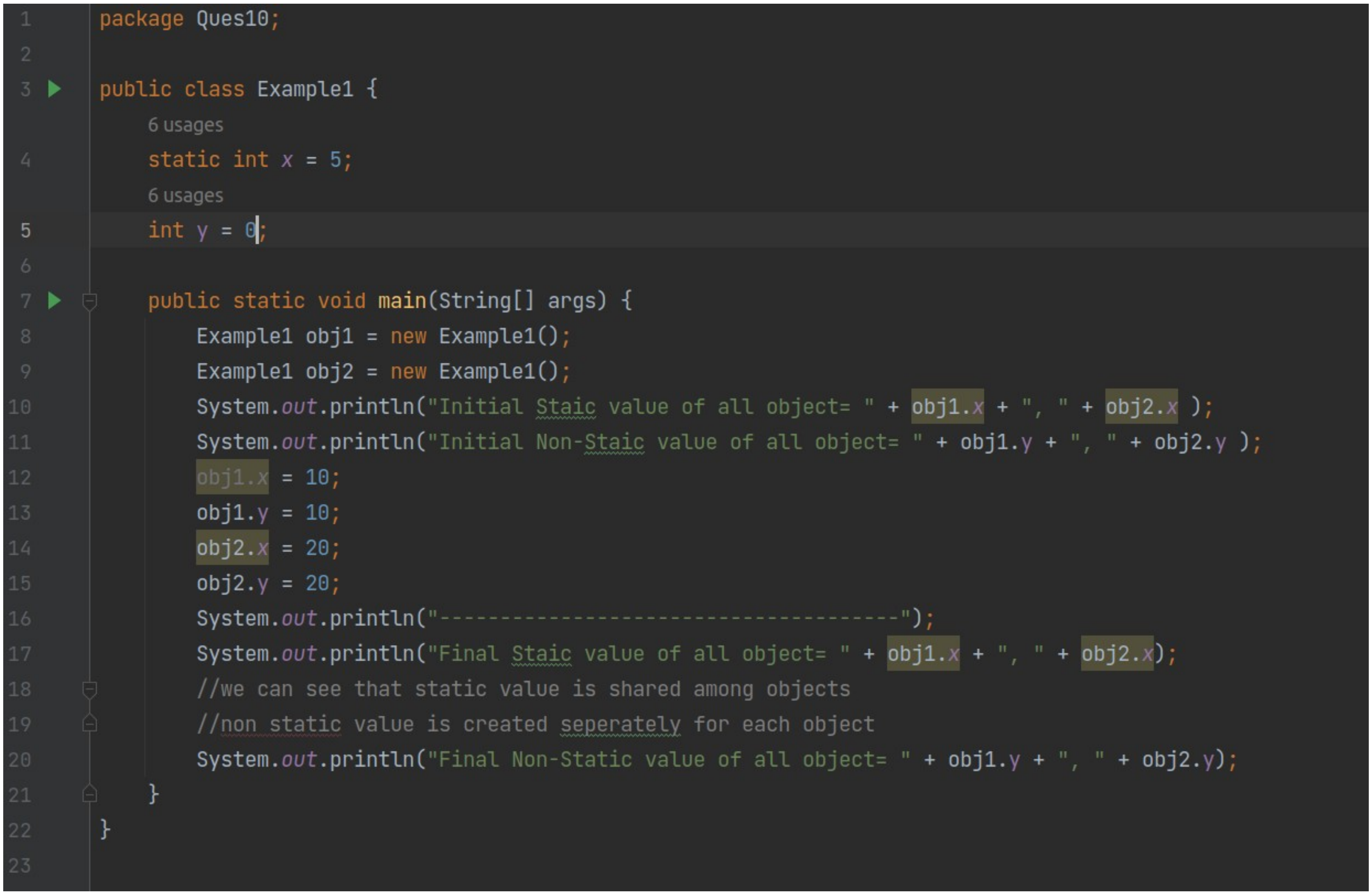


```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Radius of Circle is: 2.0
Circumference of circle is: 12.56
Area of circle is: 12.56
Diameter of circle = 4.0

Process finished with exit code 0
```

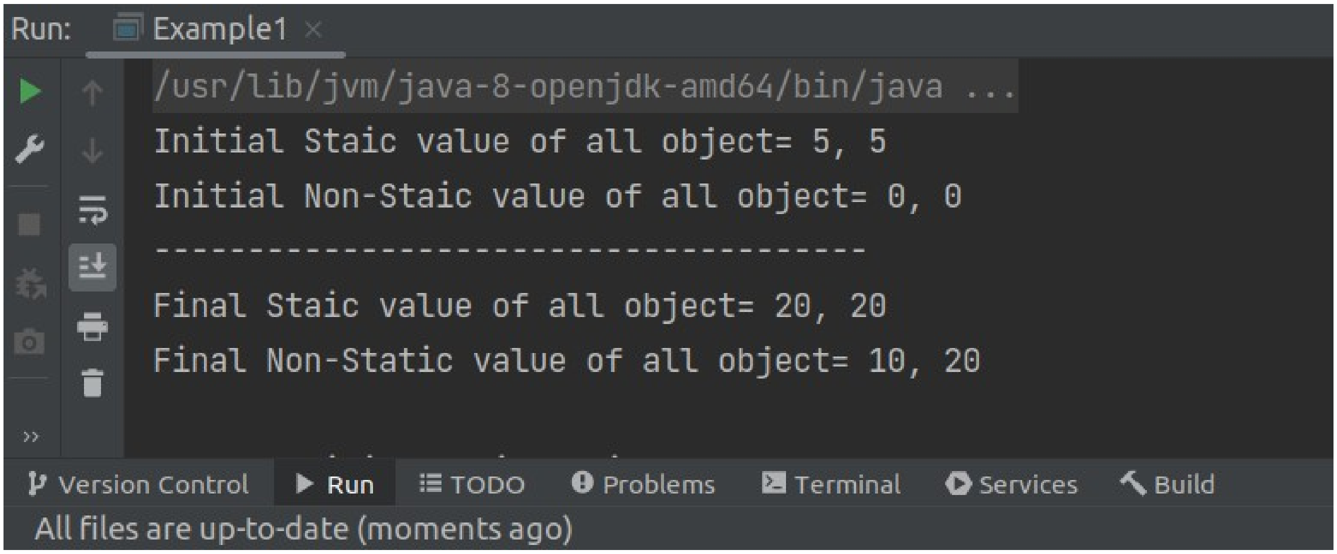
Build completed successfully in 4 sec, 603 ms (moments ago)

Ques no 10: Write 3 examples of static and non-static/ instance variable.

Ans:
<u>First Example</u>

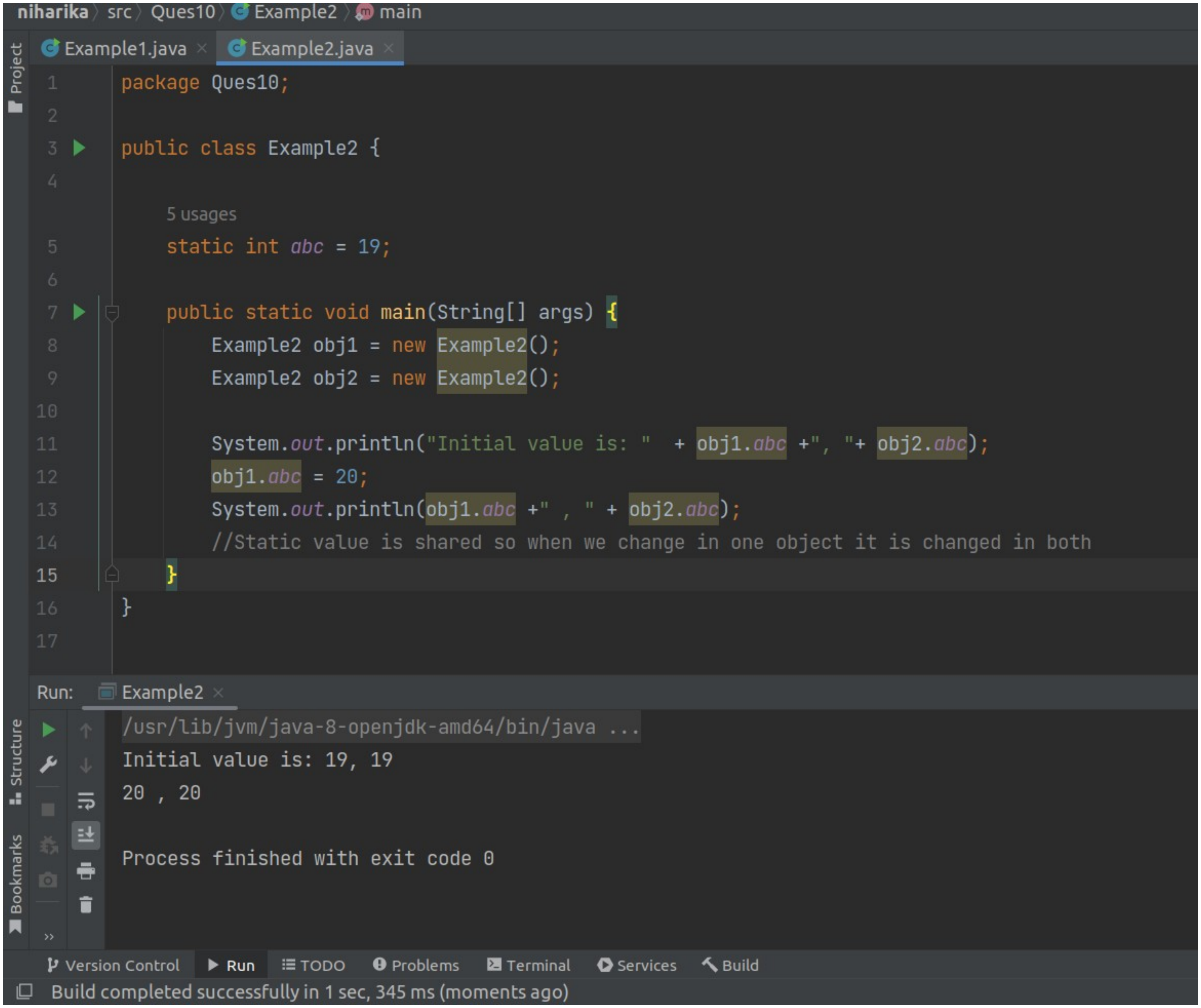


```java
package Ques10;

public class Example1 {
    static int x = 5;
    int y = 0;

    public static void main(String[] args) {
        Example1 obj1 = new Example1();
        Example1 obj2 = new Example1();
        System.out.println("Initial Staic value of all object= " + obj1.x + ", " + obj2.x );
        System.out.println("Initial Non-Staic value of all object= " + obj1.y + ", " + obj2.y );
        obj1.x = 10;
        obj1.y = 10;
        obj2.x = 20;
        obj2.y = 20;
        System.out.println("------------------------------------");
        System.out.println("Final Staic value of all object= " + obj1.x + ", " + obj2.x);
        //we can see that static value is shared among objects
        //non static value is created seperately for each object
        System.out.println("Final Non-Static value of all object= " + obj1.y + ", " + obj2.y);
    }
}
```



```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Initial Staic value of all object= 5, 5
Initial Non-Staic value of all object= 0, 0
----------------------------------------
Final Staic value of all object= 20, 20
Final Non-Static value of all object= 10, 20
```

Second Example

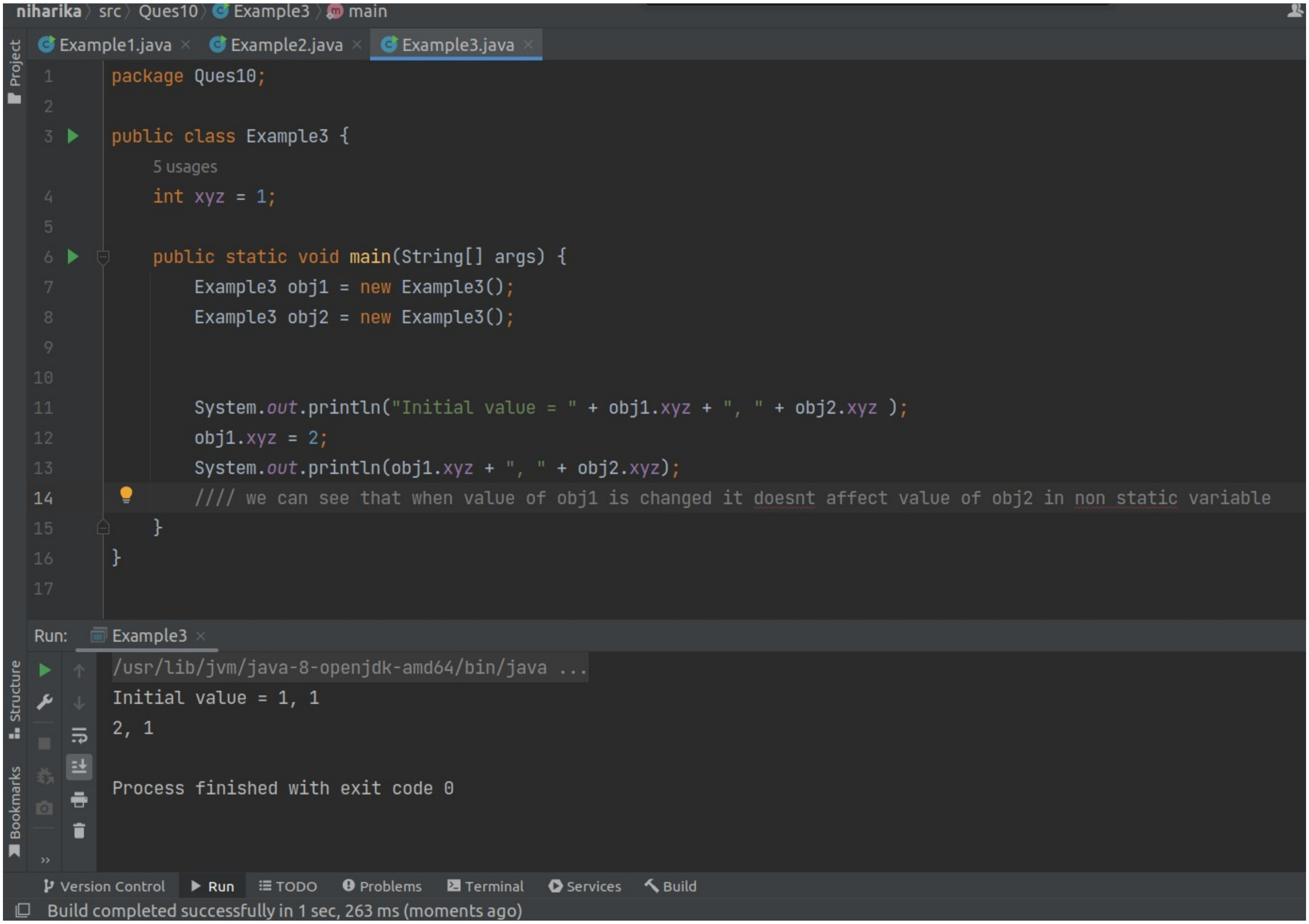


```java
package Ques10;

public class Example2 {

    static int abc = 19;

    public static void main(String[] args) {
        Example2 obj1 = new Example2();
        Example2 obj2 = new Example2();

        System.out.println("Initial value is: "  + obj1.abc +", "+ obj2.abc);
        obj1.abc = 20;
        System.out.println(obj1.abc +" , " + obj2.abc);
        //Static value is shared so when we change in one object it is changed in both
    }
}

```

```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Initial value is: 19, 19
20 , 20

Process finished with exit code 0
```

Build completed successfully in 1 sec, 345 ms (moments ago)

Third Example



```java
package Ques10;

public class Example3 {
    int xyz = 1;

    public static void main(String[] args) {
        Example3 obj1 = new Example3();
        Example3 obj2 = new Example3();


        System.out.println("Initial value = " + obj1.xyz + ", " + obj2.xyz );
        obj1.xyz = 2;
        System.out.println(obj1.xyz + ", " + obj2.xyz);
        //// we can see that when value of obj1 is changed it doesnt affect value of obj2 in non static variable
    }
}
```

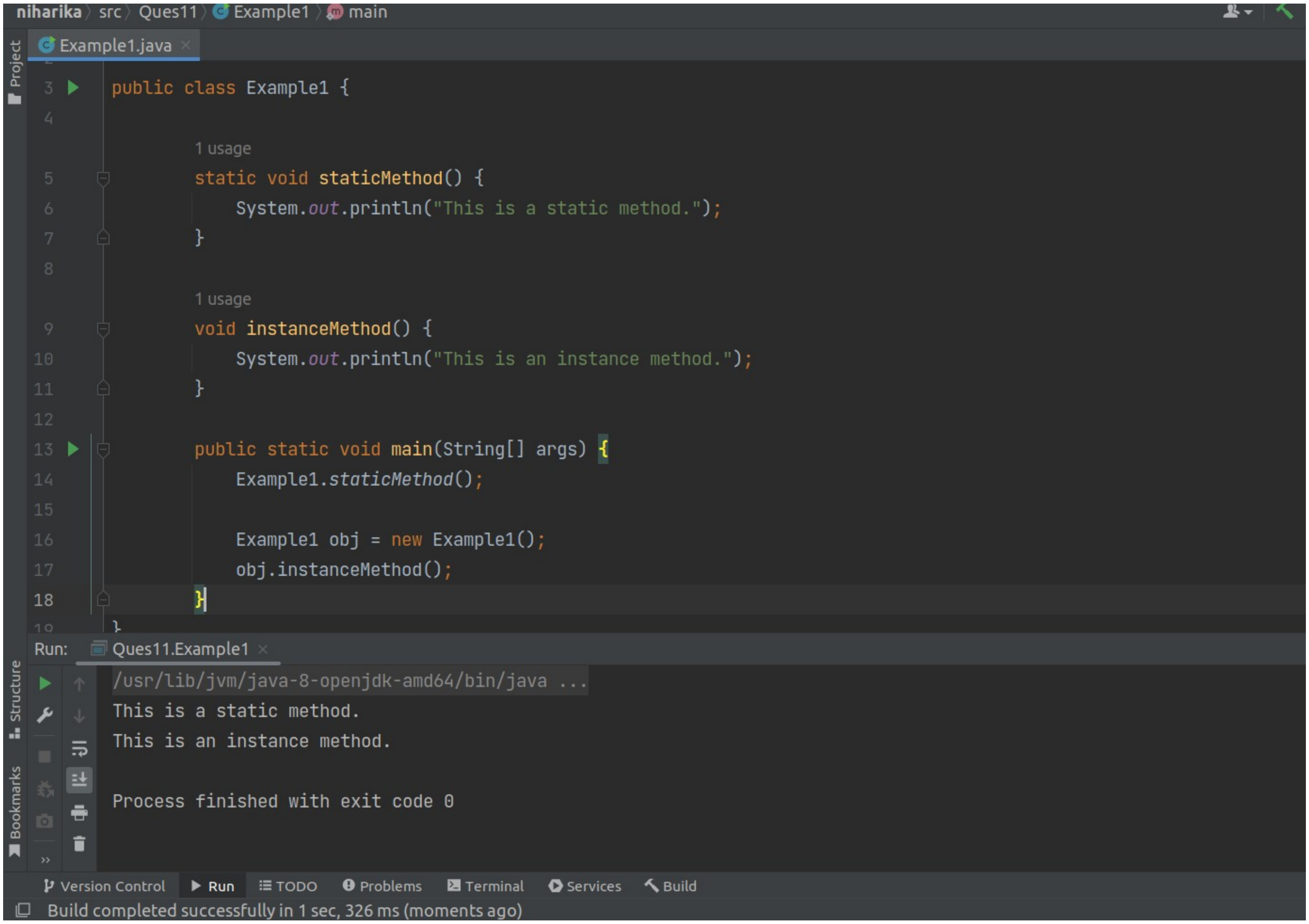
```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Initial value = 1, 1
2, 1

Process finished with exit code 0
```

Build completed successfully in 1 sec, 263 ms (moments ago)

Ques no 11. Write 3 examples of static and non-static/ instance methods.

Ans :

First Example

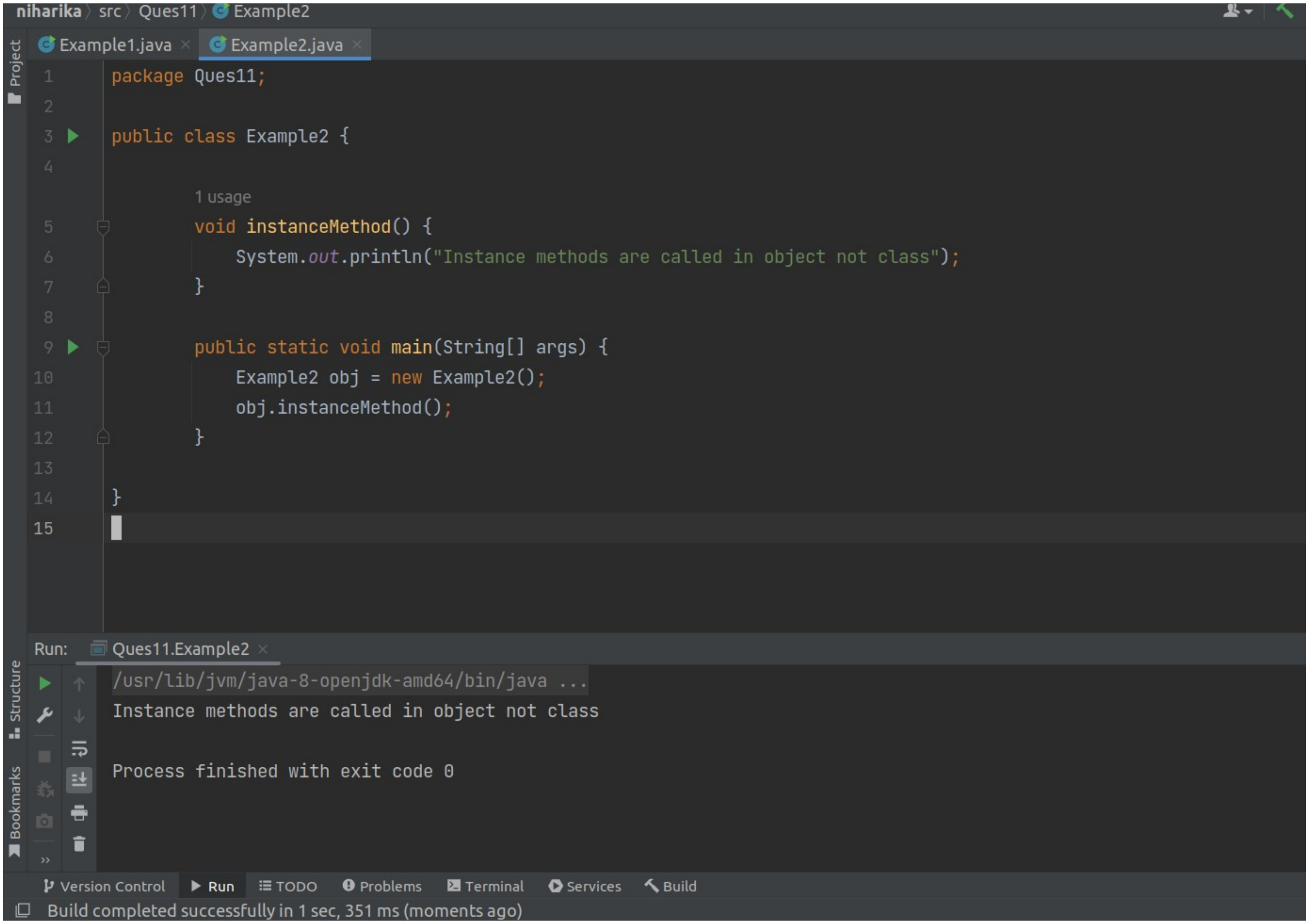


```java
public class Example1 {

        static void staticMethod() {
            System.out.println("This is a static method.");
        }

        void instanceMethod() {
            System.out.println("This is an instance method.");
        }

        public static void main(String[] args) {
            Example1.staticMethod();

            Example1 obj = new Example1();
            obj.instanceMethod();
        }
}
```

```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
This is a static method.
This is an instance method.

Process finished with exit code 0
```

Build completed successfully in 1 sec, 326 ms (moments ago)

Second Example

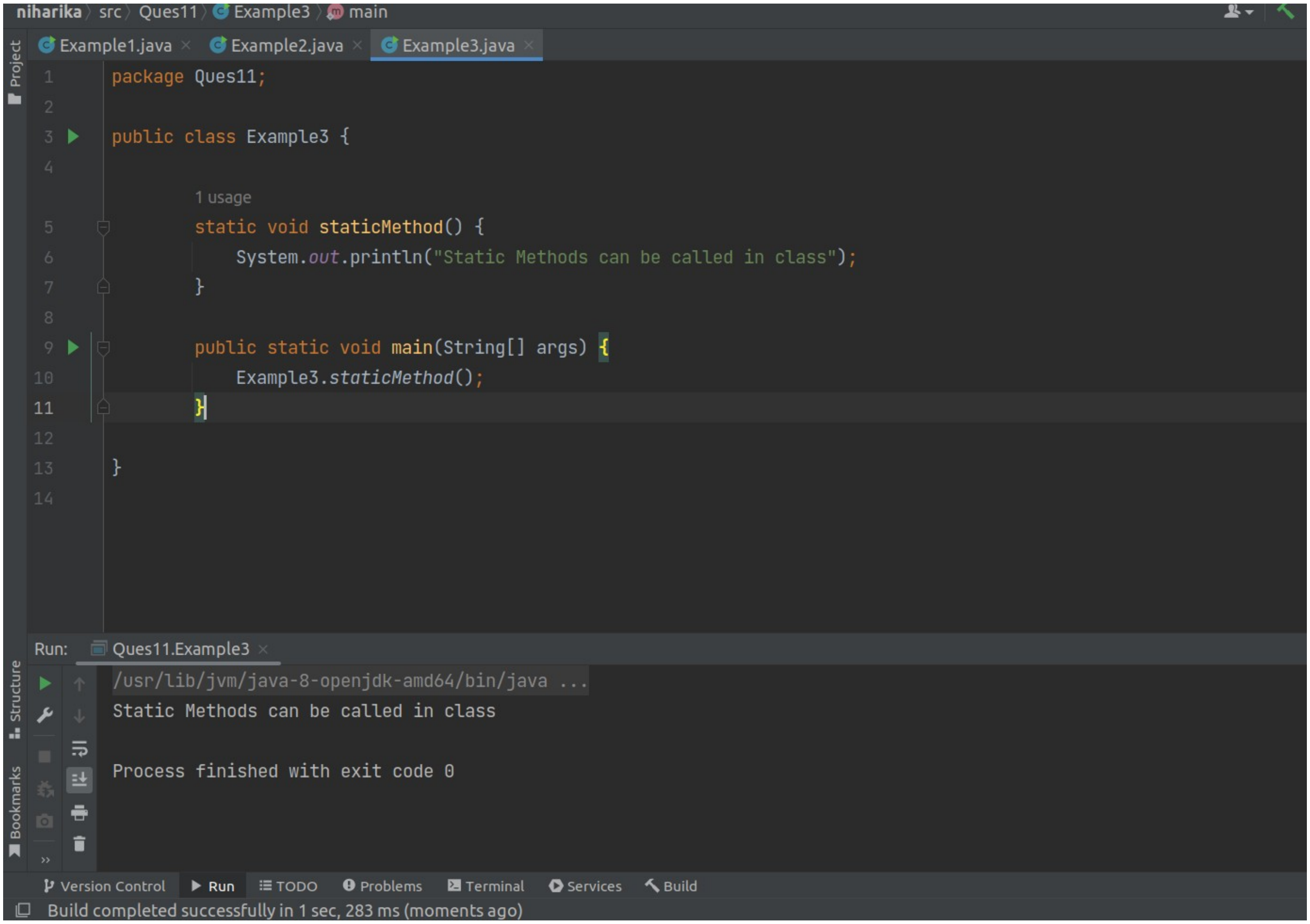


```java
package Ques11;

public class Example2 {

    1 usage
    void instanceMethod() {
        System.out.println("Instance methods are called in object not class");
    }

    public static void main(String[] args) {
        Example2 obj = new Example2();
        obj.instanceMethod();
    }

}

```

```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Instance methods are called in object not class

Process finished with exit code 0
```

Build completed successfully in 1 sec, 351 ms (moments ago)

Third Example:



```
package Ques11;

public class Example3 {

        static void staticMethod() {
            System.out.println("Static Methods can be called in class");
        }

        public static void main(String[] args) {
            Example3.staticMethod();
        }

}
```

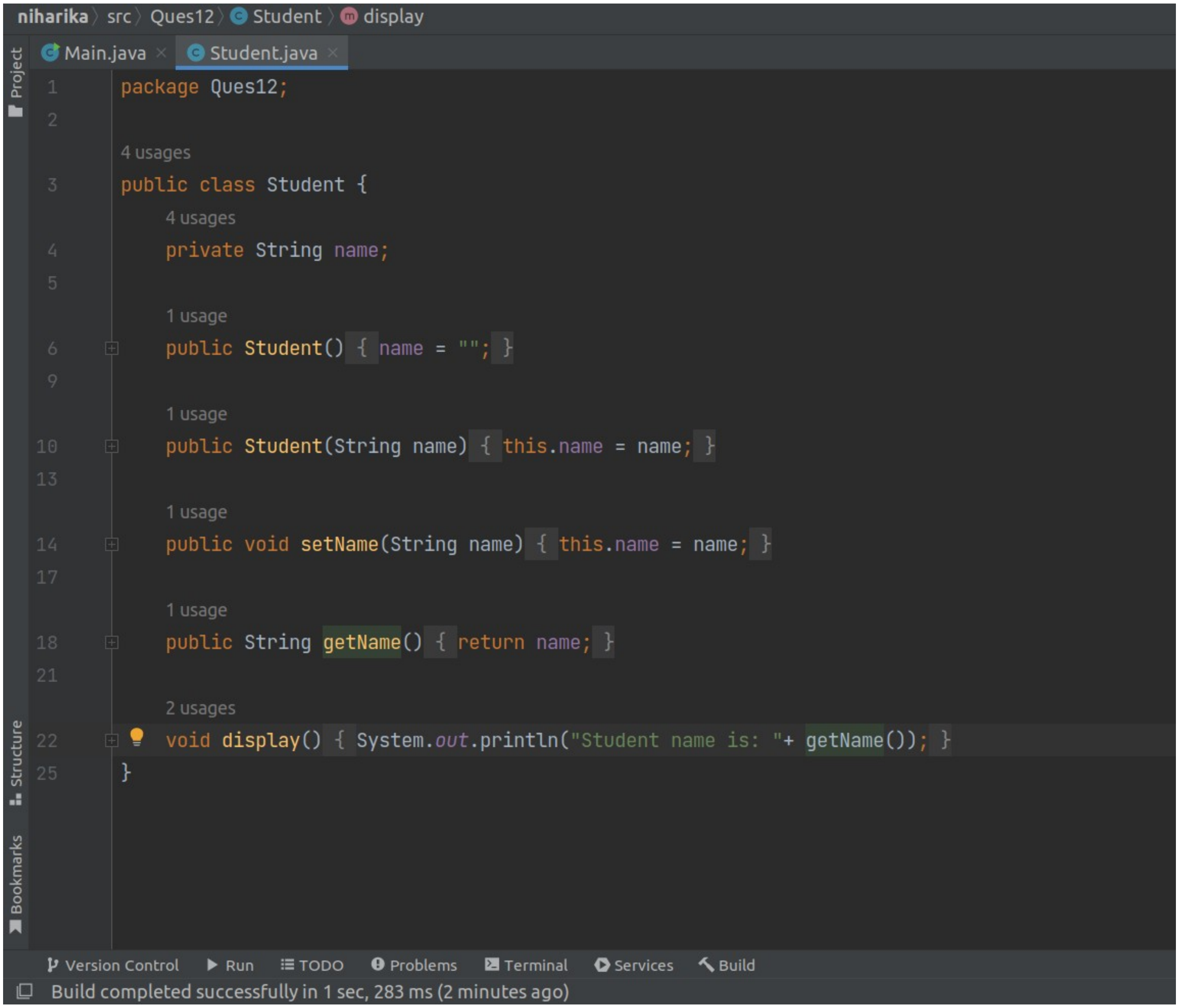
```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Static Methods can be called in class

Process finished with exit code 0
```

Build completed successfully in 1 sec, 283 ms (moments ago)

Ques no 12: Write a program of Class Student which has:
- private data member (name)
- Setter method (setName)
- Getter method (getName)

Ans:

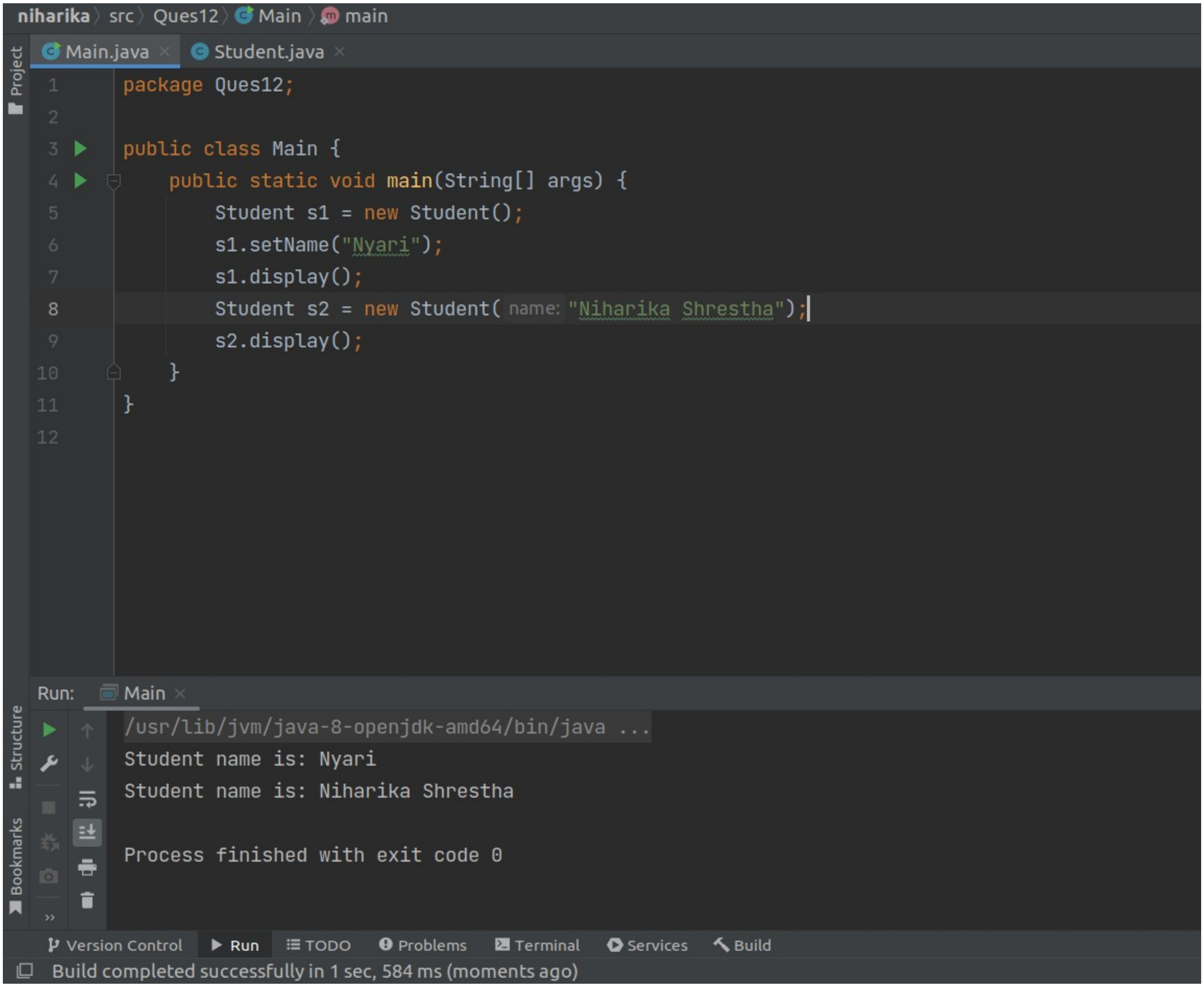
<u>Student Class File</u>



```java
package Ques12;

public class Student {
    private String name;

    public Student() { name = ""; }

    public Student(String name) { this.name = name; }

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

    public String getName() { return name; }

    void display() { System.out.println("Student name is: "+ getName()); }
}
```

Build completed successfully in 1 sec, 283 ms (2 minutes ago)

Main File + Output:



```java
package Ques12;

public class Main {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.setName("Nyari");
        s1.display();
        Student s2 = new Student( name: "Niharika Shrestha");
        s2.display();
    }
}
```

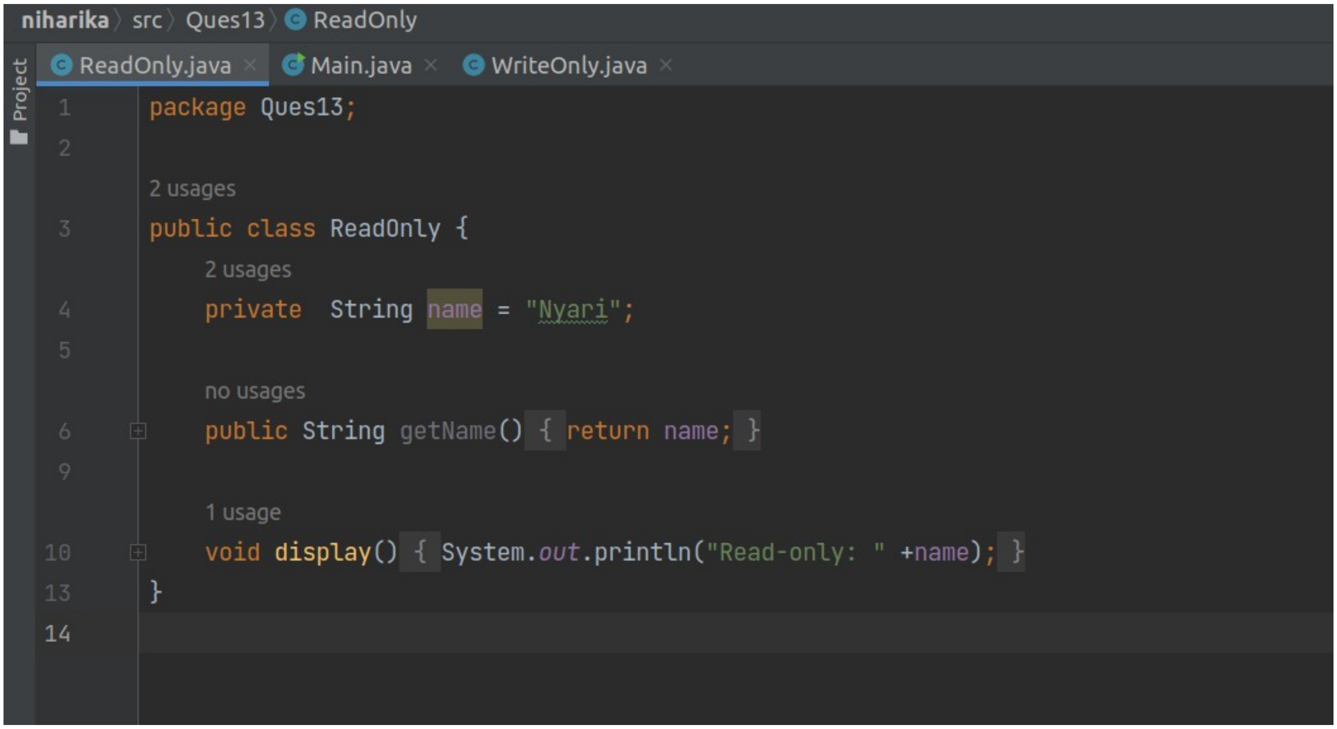
```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Student name is: Nyari
Student name is: Niharika Shrestha

Process finished with exit code 0
```

Build completed successfully in 1 sec, 584 ms (moments ago)

Ques no 13: Write an example of read-only and write-only class that has: private data member only setter method only getter method

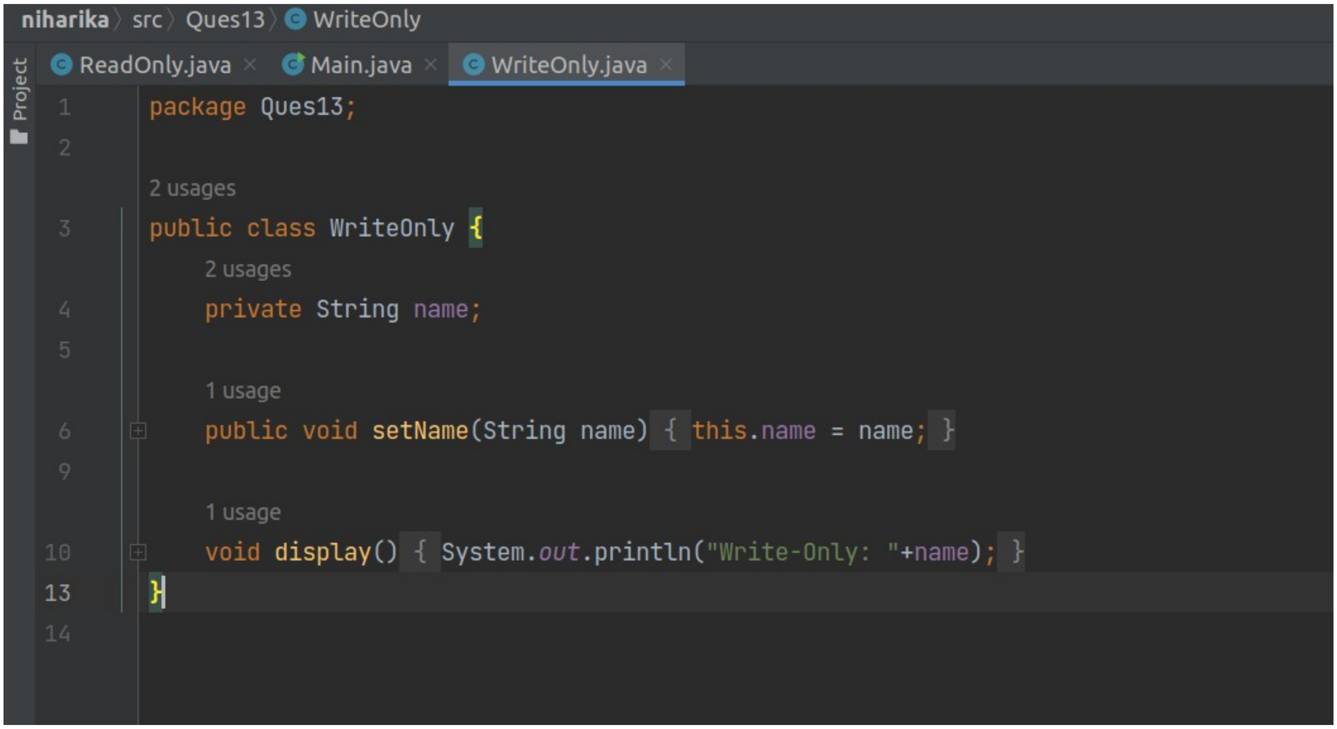
Ans:

Read-Only Class File



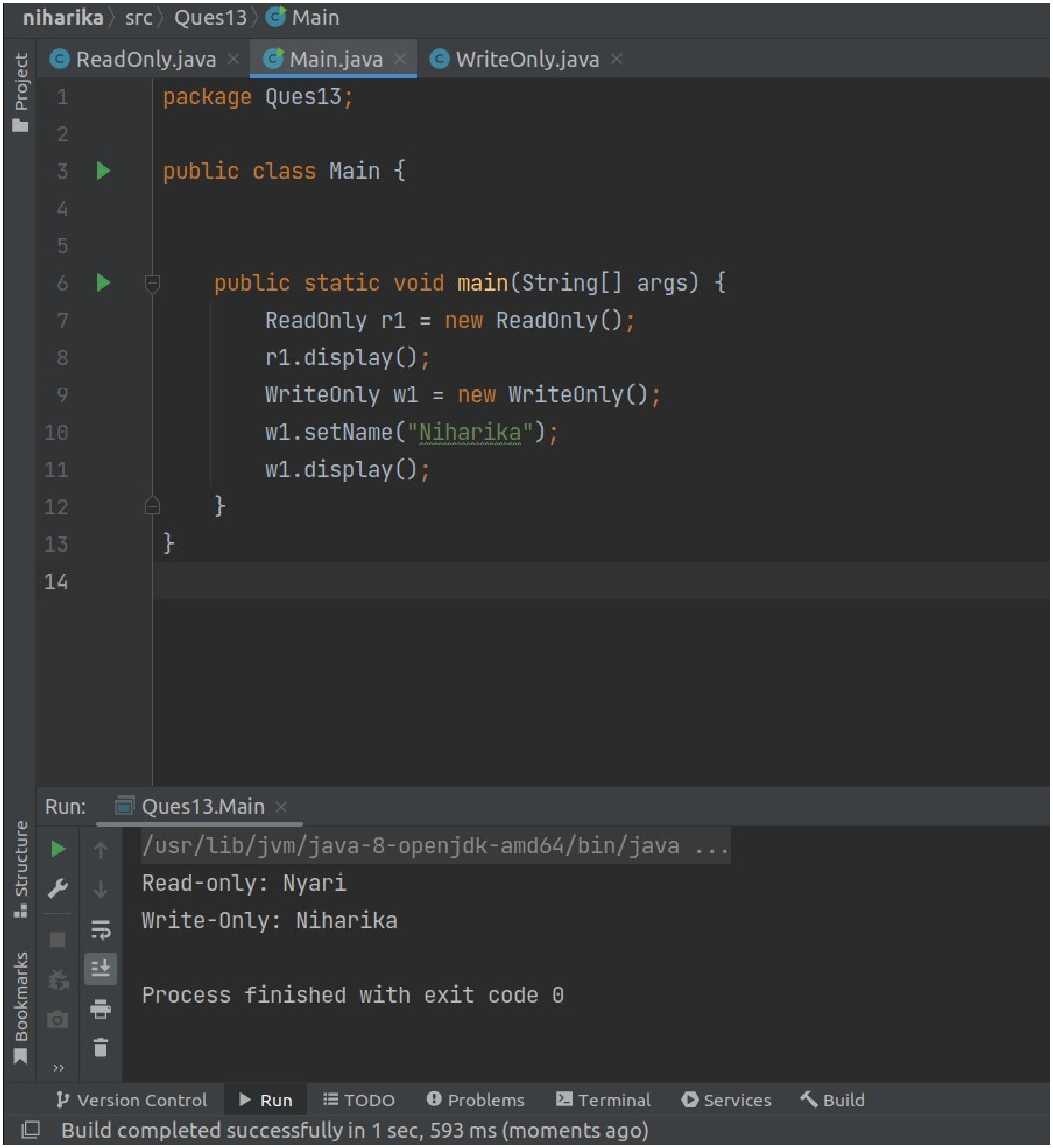
```java
package Ques13;

public class ReadOnly {
    private  String name = "Nyari";

    public String getName() { return name; }

    void display() { System.out.println("Read-only: " +name); }
}
```

Write-Only Class File



```java
package Ques13;

public class WriteOnly {
    private String name;

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

    void display() { System.out.println("Write-Only: "+name); }
}
```

Main File + Output:



```java
package Ques13;

public class Main {


    public static void main(String[] args) {
        ReadOnly r1 = new ReadOnly();
        r1.display();
        WriteOnly w1 = new WriteOnly();
        w1.setName("Niharika");
        w1.display();
    }
}
```

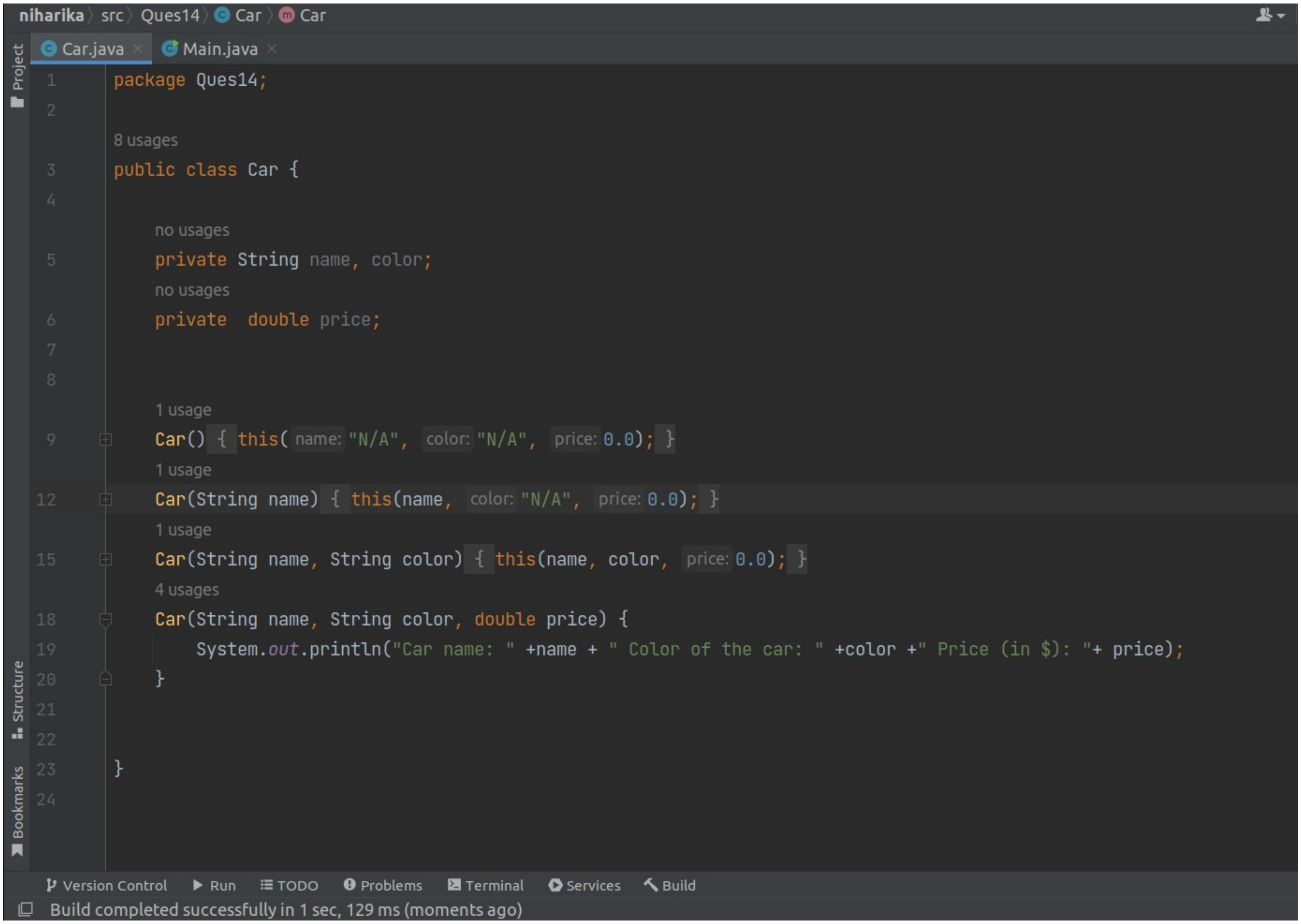
```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Read-only: Nyari
Write-Only: Niharika

Process finished with exit code 0
```

Build completed successfully in 1 sec, 593 ms (moments ago)

Q no 14: Create Class Car and create four Constructor which has:
- default constructor
- one parameterized constructor < name >
- two parameterized constructor < name and color >
- three parameterized constructor < name, color and price >
-four objects(c1, c2, c3, c4) that calls respective constructor

Ans:

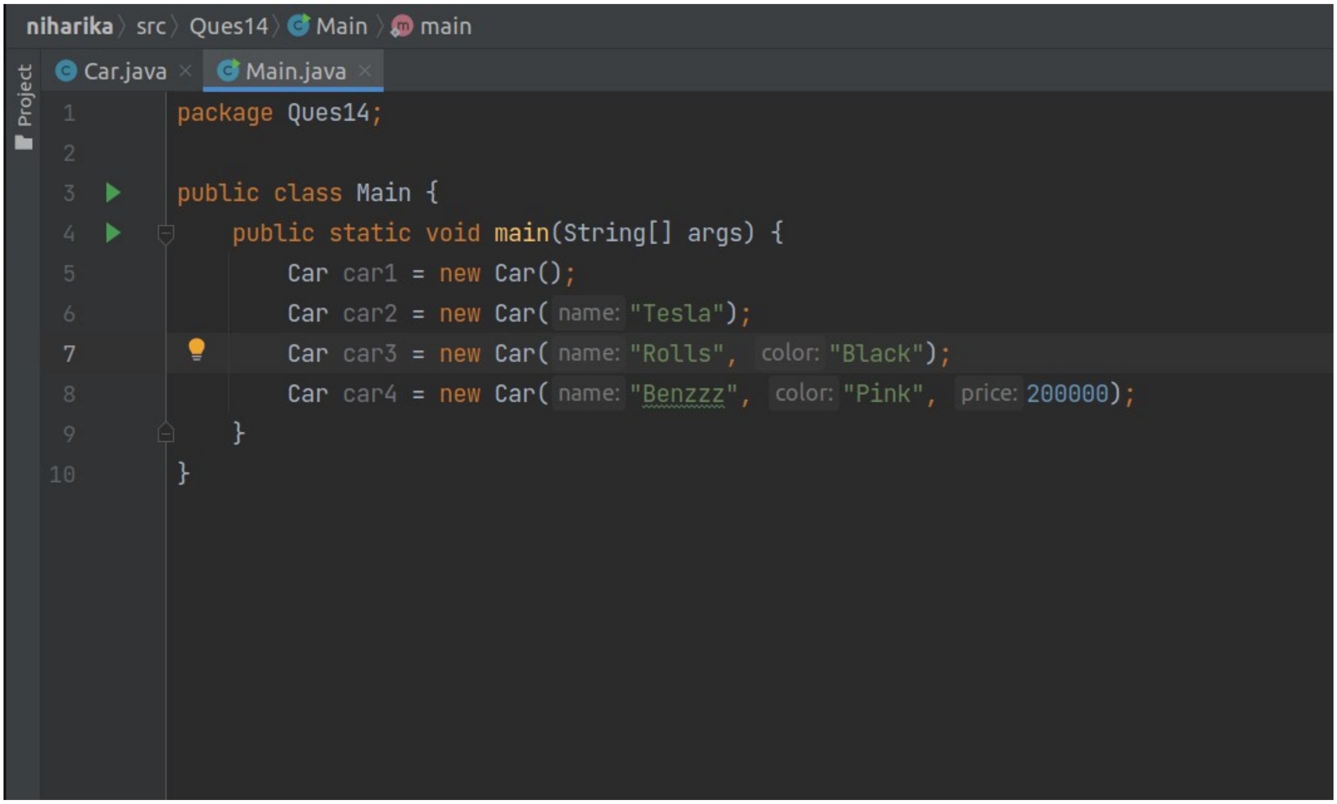
<u>Car Class File</u>



```
package Ques14;

8 usages
public class Car {

    no usages
    private String name, color;
    no usages
    private  double price;


    1 usage
    Car() { this( name: "N/A",  color: "N/A",  price: 0.0); }
    1 usage
    Car(String name) { this(name,  color: "N/A",  price: 0.0); }
    1 usage
    Car(String name, String color) { this(name, color,  price: 0.0); }
    4 usages
    Car(String name, String color, double price) {
        System.out.println("Car name: " +name + " Color of the car: " +color +" Price (in $): "+ price);
    }


}
```

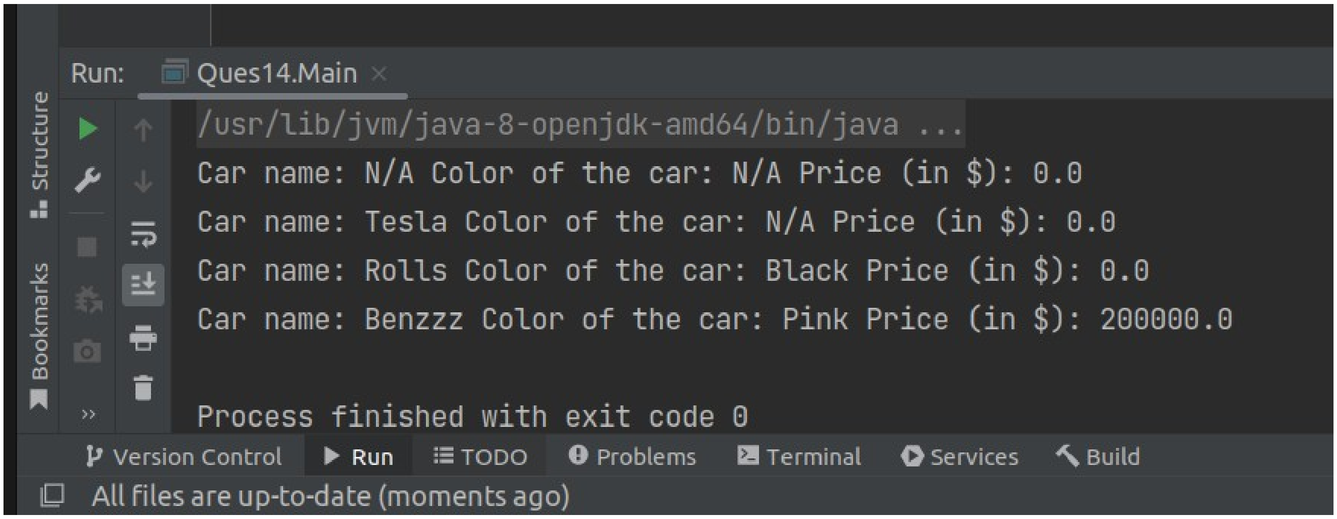
Build completed successfully in 1 sec, 129 ms (moments ago)

Main File



```java
package Ques14;

public class Main {
    public static void main(String[] args) {
        Car car1 = new Car();
        Car car2 = new Car( name: "Tesla");
        Car car3 = new Car( name: "Rolls",  color: "Black");
        Car car4 = new Car( name: "Benzzz",  color: "Pink",  price: 200000);
    }
}
```

Output



```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Car name: N/A Color of the car: N/A Price (in $): 0.0
Car name: Tesla Color of the car: N/A Price (in $): 0.0
Car name: Rolls Color of the car: Black Price (in $): 0.0
Car name: Benzzz Color of the car: Pink Price (in $): 200000.0

Process finished with exit code 0
```