

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:

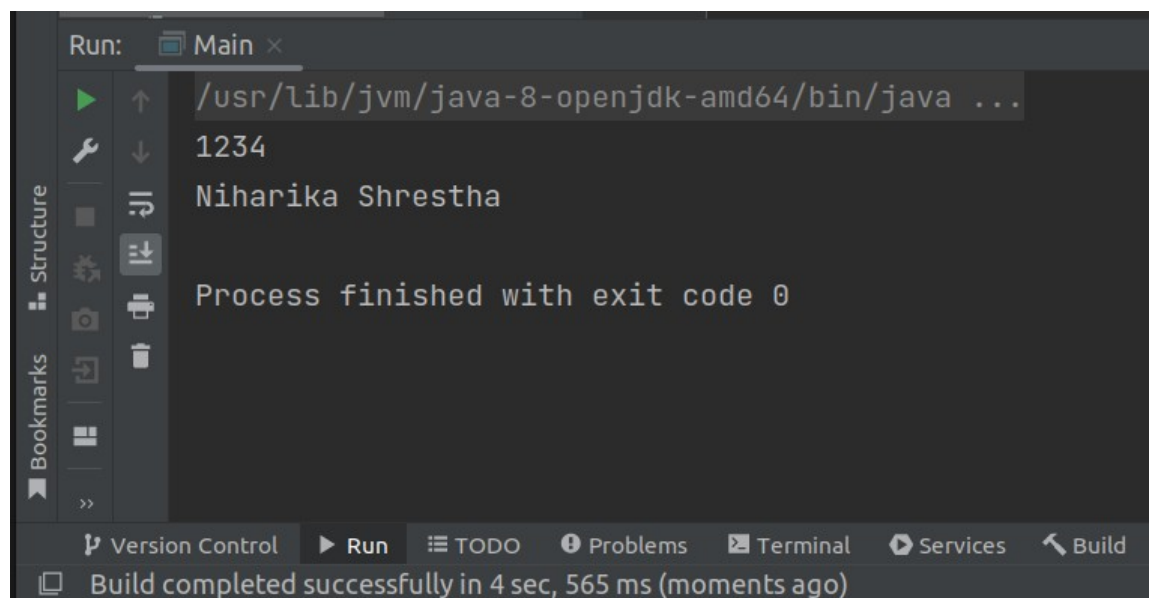
Student class

```
1 package Ques1;
2
3 public class Student {
4     int id = 1234;
5     String name = "Niharika Shrestha";
6
7     void displayName() { System.out.println(name); }
8
9
10
11    void displayId() { System.out.println(id); }
12
13
14
15 }
16
```

Main file

```
1 package Ques1;
2
3 public class Main {
4     public static void main(String[] args) {
5         Student s1 = new Student();
6         s1.displayId();
7         s1.displayName();
8     }
9 }
10
```

Output:



```
Run: Main x
/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:

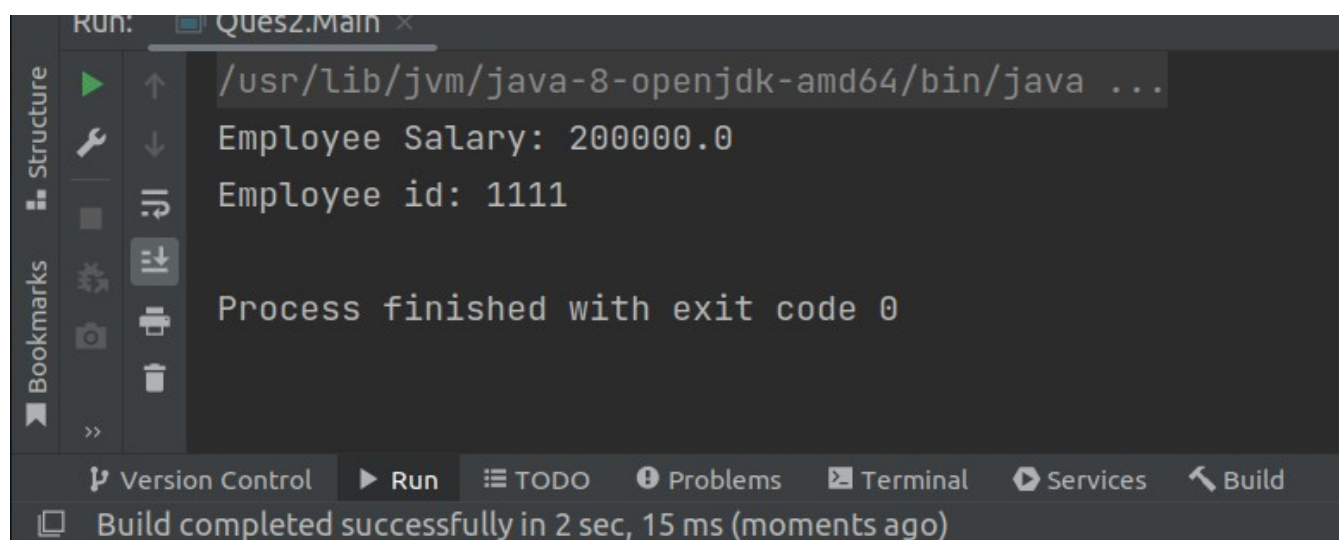
Employee file

```
1  package Ques2;
2
3  public class Employee {
4      int empId;
5      double empSalary;
6
7      void empId() {
8          System.out.println("Employee id: " + empId);
9      }
10
11     void empSalary() { System.out.println("Employee Salary: " + empSalary); }
12
13 }
14
15
```

Main file

```
1 package Ques2;
2
3
4 public class Main {
5     public static void main(String[] args) {
6         Employee emp1 = new Employee();
7         emp1.empSalary = 200000;
8         emp1.empId = 1111;
9         emp1.empSalary();
10        emp1.empId();
11    }
12 }
13
```

Output



```
Run: Ques2.Main x
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Employee Salary: 200000.0
Employee id: 1111

Process finished with exit code 0
```

Version Control Run TODO Problems Terminal Services Build

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 :

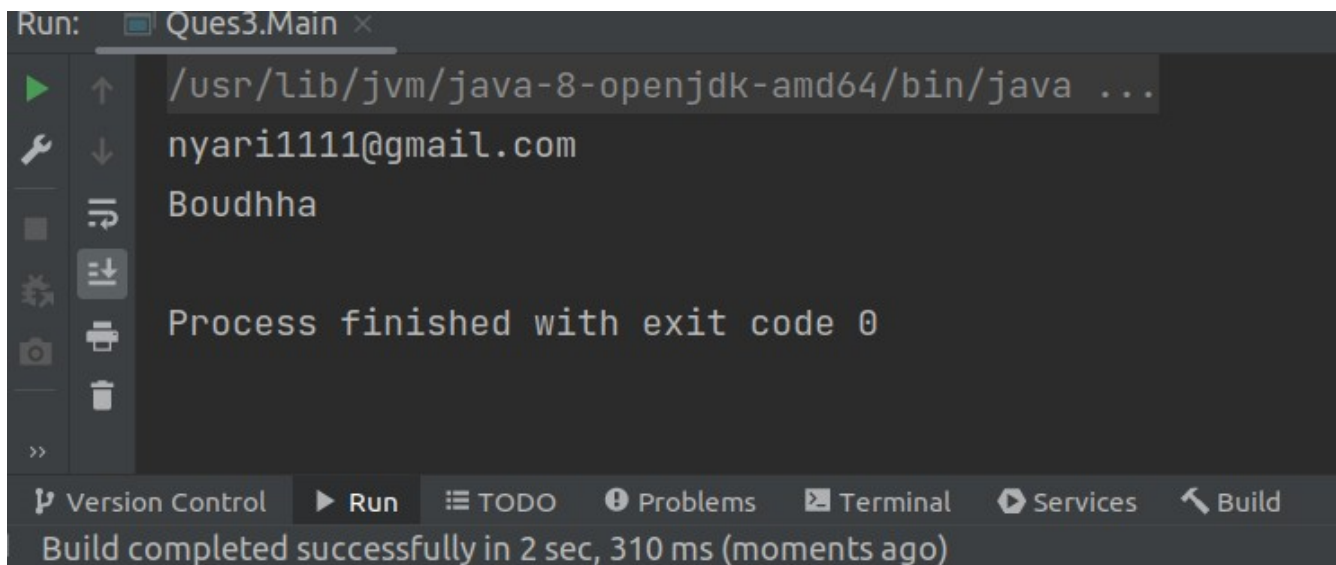
Customer File

```
1 package Ques3;
2
3 public class Customer {
4
5     2 usages
6     int customerID;
7
8     2 usages
9     String name;
10
11     2 usages
12     String address;
13
14     1 usage
15     void email() { System.out.println(name.toLowerCase()+customerID+"@gmail.com"); }
16
17     1 usage
18     void displayAddress() { System.out.println(address); }
19
20 }
```

Main file

```
1  package Ques3;
2
3  ▶ public class Main {
4  ▶  ▶ public static void main(String[] args) {
5      Customer customer1 = new Customer();
6      customer1.address = "Boudhha";
7      customer1.name = "Nyari";
8      customer1.customerID = 1111;
9      customer1.email();
10     customer1.displayAddress();
11  ▶ }
12  }
13
```

Output



```
Run: Ques3.Main x
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
nyari1111@gmail.com
Boudhha
Process finished with exit code 0

Version Control Run TODO Problems Terminal Services Build
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:

Car File

```
1 package Ques4;
2
3 public class Car {
4     private String name;
5     private String color;
6
7     private double price;
8
9     public void setName(String name) { this.name = name; }
10
11     public void setColor(String color) { this.color = color; }
12
13     public void setPrice(int price) { this.price = price; }
14
15     void display() {
16         System.out.println("Car name is: " + name );
17         System.out.println("Price is $: " + price );
18         System.out.println("Color: " + color);
19     }
20 }
```


Main file

```
1 package Ques4;
2
3
4 public class Main {
5     public static void main(String[] args) {
6         Car c1 = new Car();
7         c1.setName("Benz");
8         c1.setColor("Pink");
9         c1.setPrice(4500000);
10        c1.display();
11    }
12 }
13
```

Output:

Run: Ques4.Main x

```
/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
```

Version Control Run TODO Problems Terminal Services Build

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:

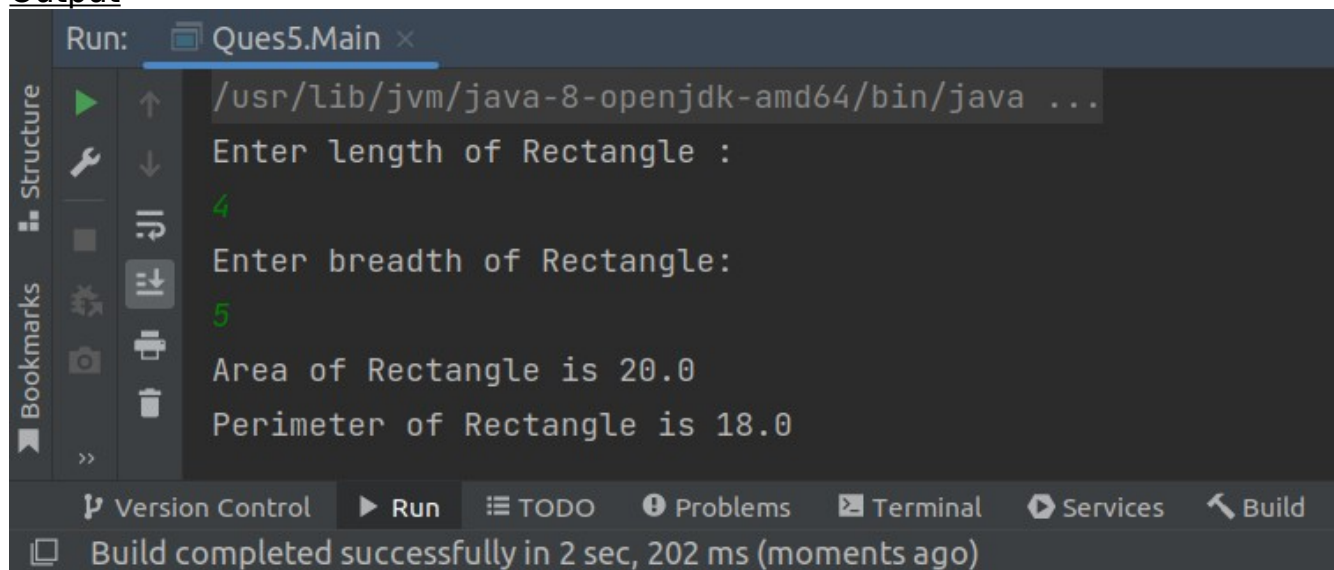
Rectangle Class file

```
1 package Ques5;
2
3 public class Rectangle {
4     private float length;
5
6     private float breadth;
7
8     public void setLength(float length) { this.length = length; }
9
10    public float getLength() { return length; }
11
12    public float getBreadth() { return breadth; }
13
14    public void setBreadth(float breadth) { this.breadth = breadth; }
15
16    void display() {
17        System.out.println("Area of Rectangle is " + (length*breadth));
18        System.out.println("Perimeter of Rectangle is " + 2*(length+breadth));
19    }
20 }
```

Main File

```
1 package Ques5;|
2
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args) {
7         Rectangle r1 = new Rectangle();
8         Scanner scanner = new Scanner(System.in);
9         System.out.println("Enter length of Rectangle : ");
10        r1.setLength(scanner.nextFloat());
11        System.out.println("Enter breadth of Rectangle: ");
12        r1.setBreadth(scanner.nextFloat());
13        scanner.close();
14
15        r1.display();
16    }
17 }
18
```

Output



```
Run: Ques5.Main x
/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:

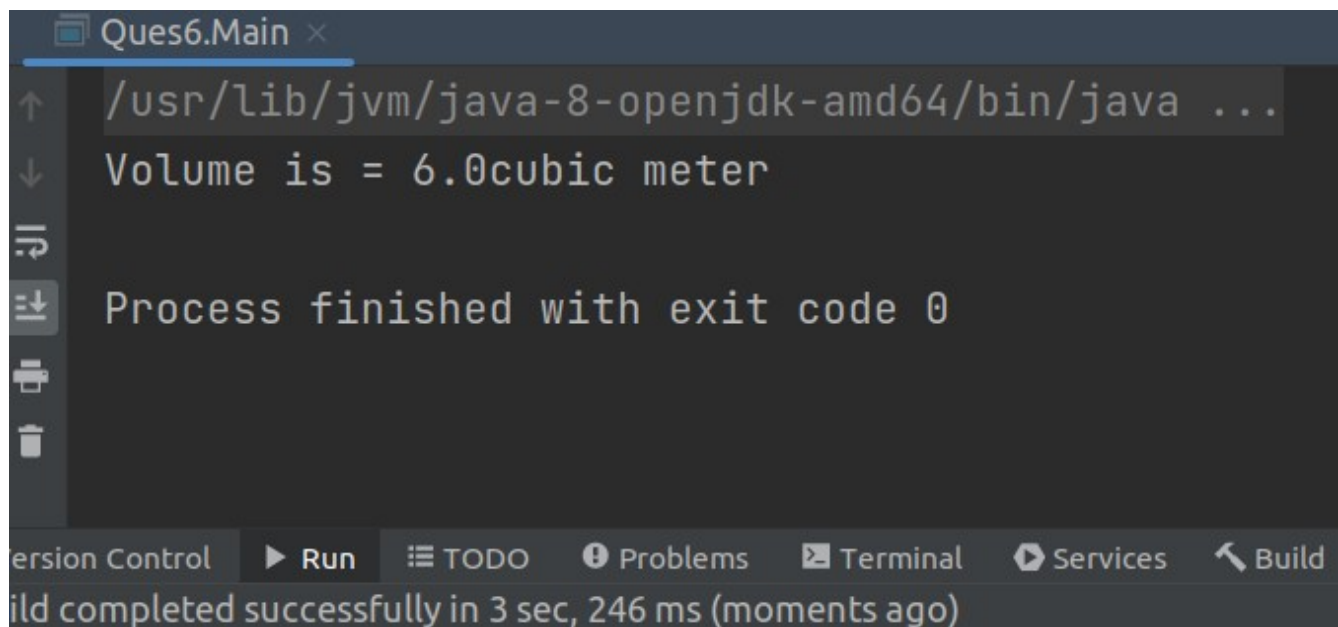
Cuboid Class File

```
1  package Ques6;
2
3  2 usages
4  public class Cuboid {
5      3 usages
6      private float length;
7      3 usages
8      private float breadth;
9      3 usages
10     private float height;
11
12     no usages
13     public float getLength() { return length; }
14
15     1 usage
16     public void setLength(float length) { this.length = length; }
17
18     no usages
19     public float getBreadth() { return breadth; }
20
21     1 usage
22     public void setBreadth(float breadth) { this.breadth = breadth; }
23
24     no usages
25     public float getHeight() { return height; }
26
27     1 usage
28     public void setHeight(float height) { this.height = height; }
29
30     1 usage
31     void display() {
32         System.out.println("Volume is =" + length*breadth*height + "cubic meter");
33     }
34 }
```

Main file

```
1 package Ques6;  
2  
3 public class Main {  
4     public static void main(String[] args) {  
5         Cuboid c1 = new Cuboid();  
6         c1.setLength(1);  
7         c1.setBreadth(2);  
8         c1.setHeight(3);  
9         c1.display();  
10    }  
11 }  
12 |
```

Output:



```
Ques6.Main x  
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...  
Volume is = 6.0cubic meter  
Process finished with exit code 0  
Version Control Run TODO Problems Terminal Services Build  
Build 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:

Interest Class File

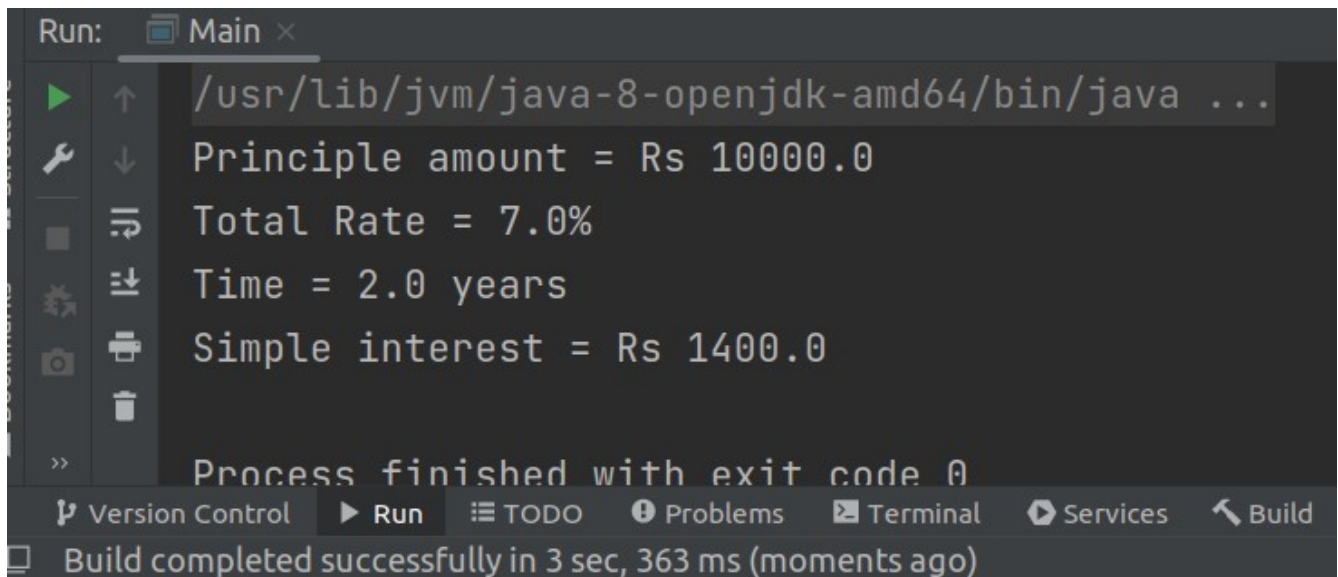
```
3      public class Interest {  
4          3 usages  
5          private double principle;  
6          3 usages  
7          private double time;  
8          3 usages  
9          private double rate;  
10  
11      1 usage  
12      public Interest(double principle, double time, double rate) {  
13          this.principle = principle;  
14          this.rate = rate;  
15          this.time = time;  
16      }  
17  
18      public void display() {  
19          System.out.println("Principle amount = Rs " + principle);  
20          System.out.println("Total Rate = " + rate + "%");  
21          System.out.println("Time = " + time + " years");  
22      }  
23  
24      1 usage  
25      void calculate() {  
26          double si;  
27          si = (principle * rate * time) / 100;  
28          System.out.println("Simple interest = Rs " + si);  
29      }  
30  }
```

Problems Terminal Services Build

Main file

```
1 package Ques7;
2
3 public class Main {
4     public static void main(String[] args) {
5         Interest interest1 = new Interest( principle: 10000, time: 2, rate: 7);
6         interest1.display();
7         interest1.calculate();
8     }
9 }
10
```

Output:



Run: Main x

```
/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
```

Version Control Run TODO Problems Terminal Services Build

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

```
2 usages
3  public class Account {
4      2 usages
5      private String name;
6      7 usages
7      private double amount;
8
9      1 usage
10     public Account(String name, double amount) {
11         this.amount = amount;
12         this.name = name;
13     }
14     1 usage
15     public void withdrawAmount(double withdraw) {
16         if(withdraw < amount) {
17             amount -= withdraw;
18             System.out.println("Balance after Withdraw of Rs."+withdraw+" is Rs."+amount);
19         } else {
20             System.out.println("Insufficient amount");
21         }
22     }
23
24     1 usage
25     public void depositAmount(double deposit) {
26         amount += deposit;
27         System.out.println("Balance after Deposit of Rs."+deposit+" is Rs."+amount);
28     }
29     public void display() {
30         System.out.println("Account name: " + name);
31         System.out.println("Account amount: " + amount);
32     }
33 }
```

Version Control Run TODO Problems Terminal Services Build

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

Main File

```
1 package Ques8;
2
3
4 public class Main {
5     public static void main(String[] args) {
6         Account account1 = new Account( name: "Niharika Shrestha", amount: 1000000);
7         account1.display();
8         account1.withDrawAmount(10000);
9         account1.depositAmount(500);
10    }
11 }
12
```

Output:

Run: Ques8.Main x

```
/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
```

Version Control Run TODO Problems Terminal Services Build

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

```
1  package Ques9;
2
3  public class Circle {
4      private double radius;
5
6      public Circle(double radius){
7          this.radius= radius;
8          System.out.println("Radius of Circle is: " + radius);
9      }
10
11     public void circumference() {
12         double c = 2*3.14*radius;
13         System.out.println("Circumference of circle is: " + c);
14     }
15
16     public void area() {
17         double area = 3.14*radius*radius;
18         System.out.println("Area of circle is: " + area);
19     }
20
21     public void diameter() { System.out.println("Diameter of circle = " + (radius+radius)); }
22
23 }
24
```

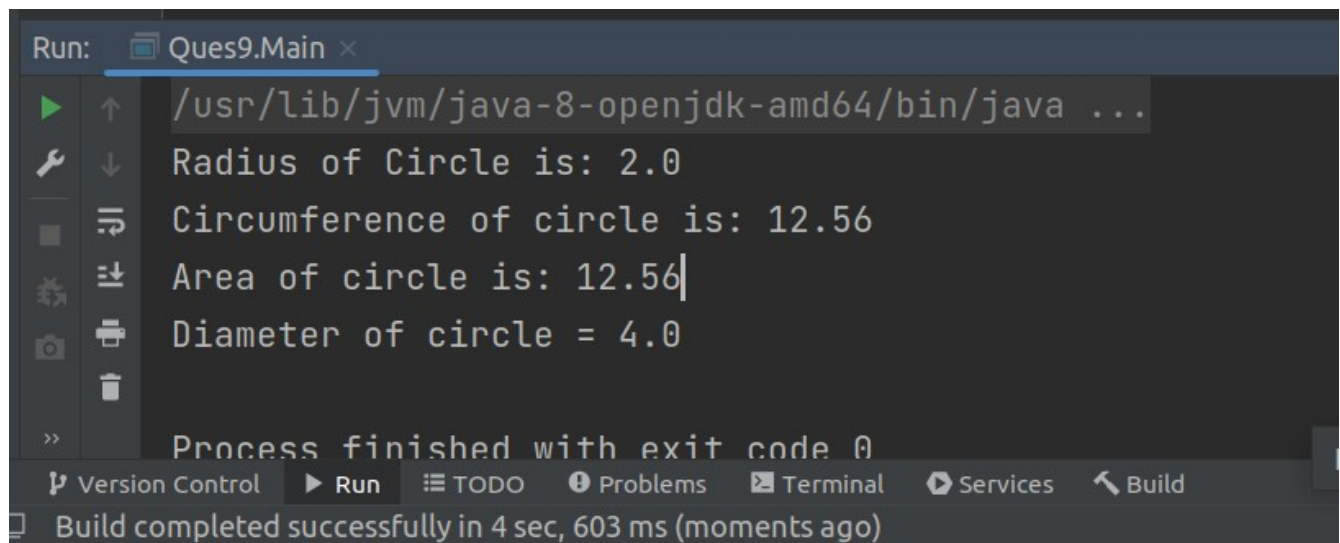
Version Control Run TODO Problems Terminal Services Build

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

Main File

```
1 package Ques9;
2
3
4 public class Main {
5     public static void main(String[] args) {
6         Circle c1 = new Circle( radius: 2);
7         c1.circumference();
8         c1.area();
9         c1.diameter();
10    }
11 }
12
```

Output



```
Run: Ques9.Main x
/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
Version Control Run TODO Problems Terminal Services Build
Build completed successfully in 4 sec, 603 ms (moments ago)
```

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

Ans:

First Example

```
1 package Ques10;
2
3 public class Example1 {
4     static int x = 5;
5     int y = 0;
6
7     public static void main(String[] args) {
8         Example1 obj1 = new Example1();
9         Example1 obj2 = new Example1();
10        System.out.println("Initial Staic value of all object= " + obj1.x + ", " + obj2.x );
11        System.out.println("Initial Non-Staic value of all object= " + obj1.y + ", " + obj2.y );
12        obj1.x = 10;
13        obj1.y = 10;
14        obj2.x = 20;
15        obj2.y = 20;
16        System.out.println("-----");
17        System.out.println("Final Staic value of all object= " + obj1.x + ", " + obj2.x);
18        //we can see that static value is shared among objects
19        //non static value is created seperately for each object
20        System.out.println("Final Non-Static value of all object= " + obj1.y + ", " + obj2.y);
21    }
22 }
23
```

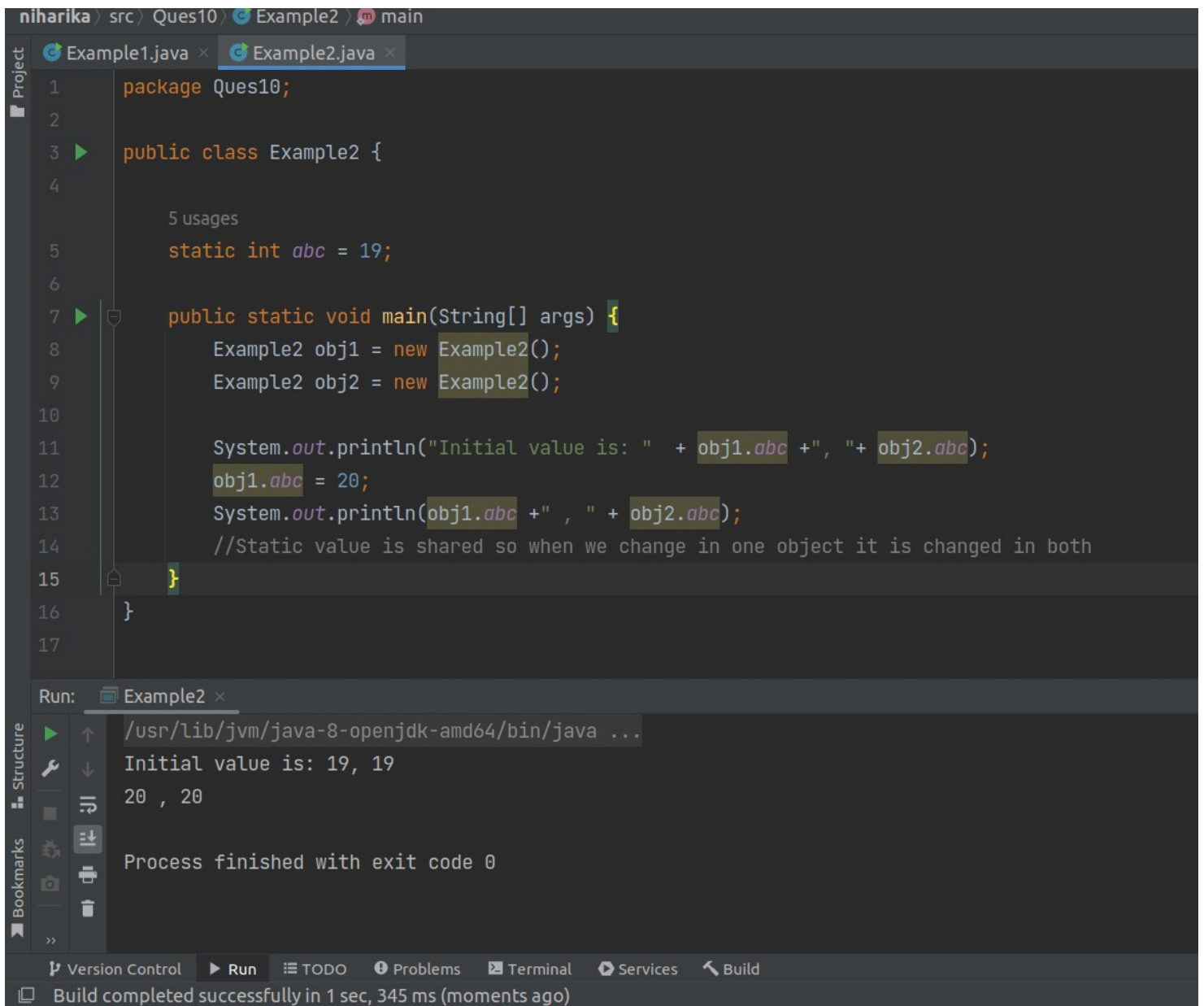
Run: Example1 x

```
/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
```

Version Control Run TODO Problems Terminal Services Build

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

Second Example



The screenshot shows an IDE with two tabs: Example1.java and Example2.java. The Example2.java tab is active, displaying the following code:

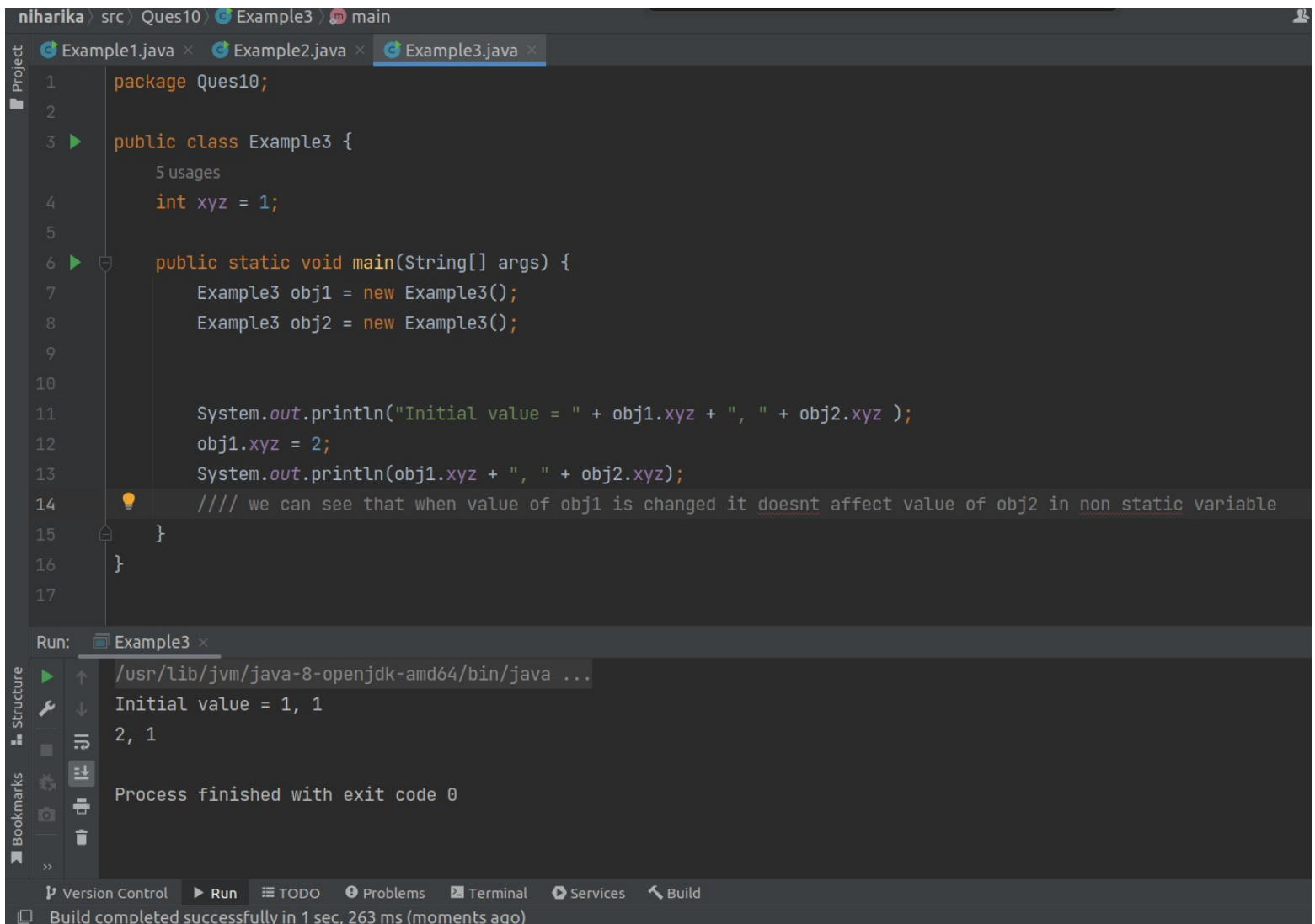
```
1 package Ques10;
2
3 public class Example2 {
4
5     5 usages
6     static int abc = 19;
7
8     public static void main(String[] args) {
9         Example2 obj1 = new Example2();
10        Example2 obj2 = new Example2();
11
12        System.out.println("Initial value is: " + obj1.abc + ", " + obj2.abc);
13        obj1.abc = 20;
14        System.out.println(obj1.abc + " , " + obj2.abc);
15        //Static value is shared so when we change in one object it is changed in both
16    }
17 }
```

Below the code editor, the Run console shows the execution of Example2. The output is:

```
Run: Example2 x
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Initial value is: 19, 19
20 , 20
Process finished with exit code 0
```

The bottom status bar indicates: Build completed successfully in 1 sec, 345 ms (moments ago).

Third Example



The screenshot shows an IDE with the following components:

- Project View:** Shows the file structure with `Example1.java`, `Example2.java`, and `Example3.java`.
- Editor:** Displays the code for `Example3.java`. The code is as follows:

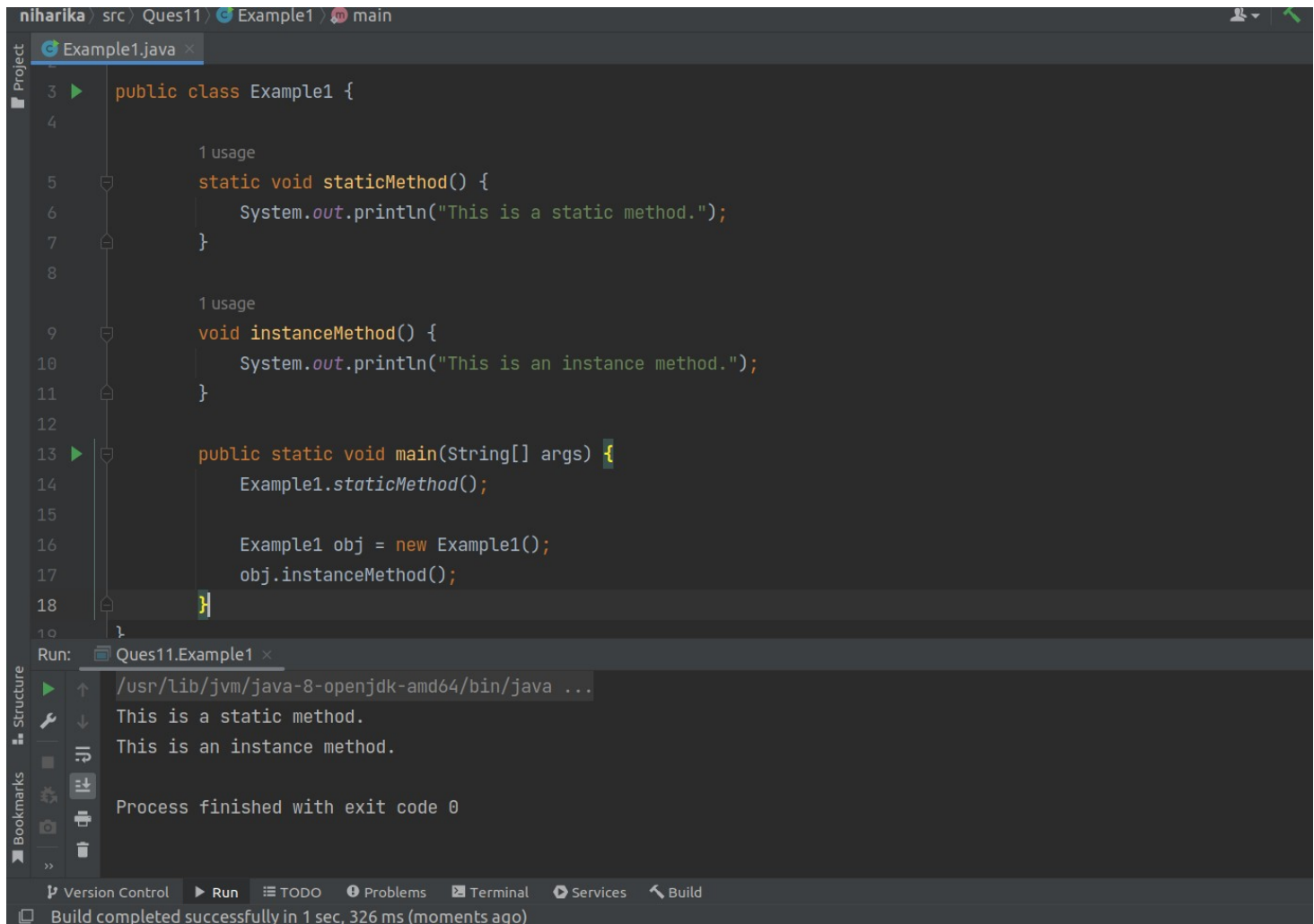
```
1 package Ques10;
2
3 public class Example3 {
4     5 usages
5     int xyz = 1;
6
7     public static void main(String[] args) {
8         Example3 obj1 = new Example3();
9         Example3 obj2 = new Example3();
10
11         System.out.println("Initial value = " + obj1.xyz + ", " + obj2.xyz );
12         obj1.xyz = 2;
13         System.out.println(obj1.xyz + ", " + obj2.xyz);
14         /// we can see that when value of obj1 is changed it doesnt affect value of obj2 in non static variable
15     }
16 }
17
```
- Run View:** Shows the execution output for `Example3`. The output is:

```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Initial value = 1, 1
2, 1
Process finished with exit code 0
```
- Bottom Bar:** Contains tabs for `Version Control`, `Run`, `TODO`, `Problems`, `Terminal`, `Services`, and `Build`. A status bar at the bottom indicates "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



The screenshot shows an IDE with a project named 'Ques11'. The main file is 'Example1.java', which contains the following code:

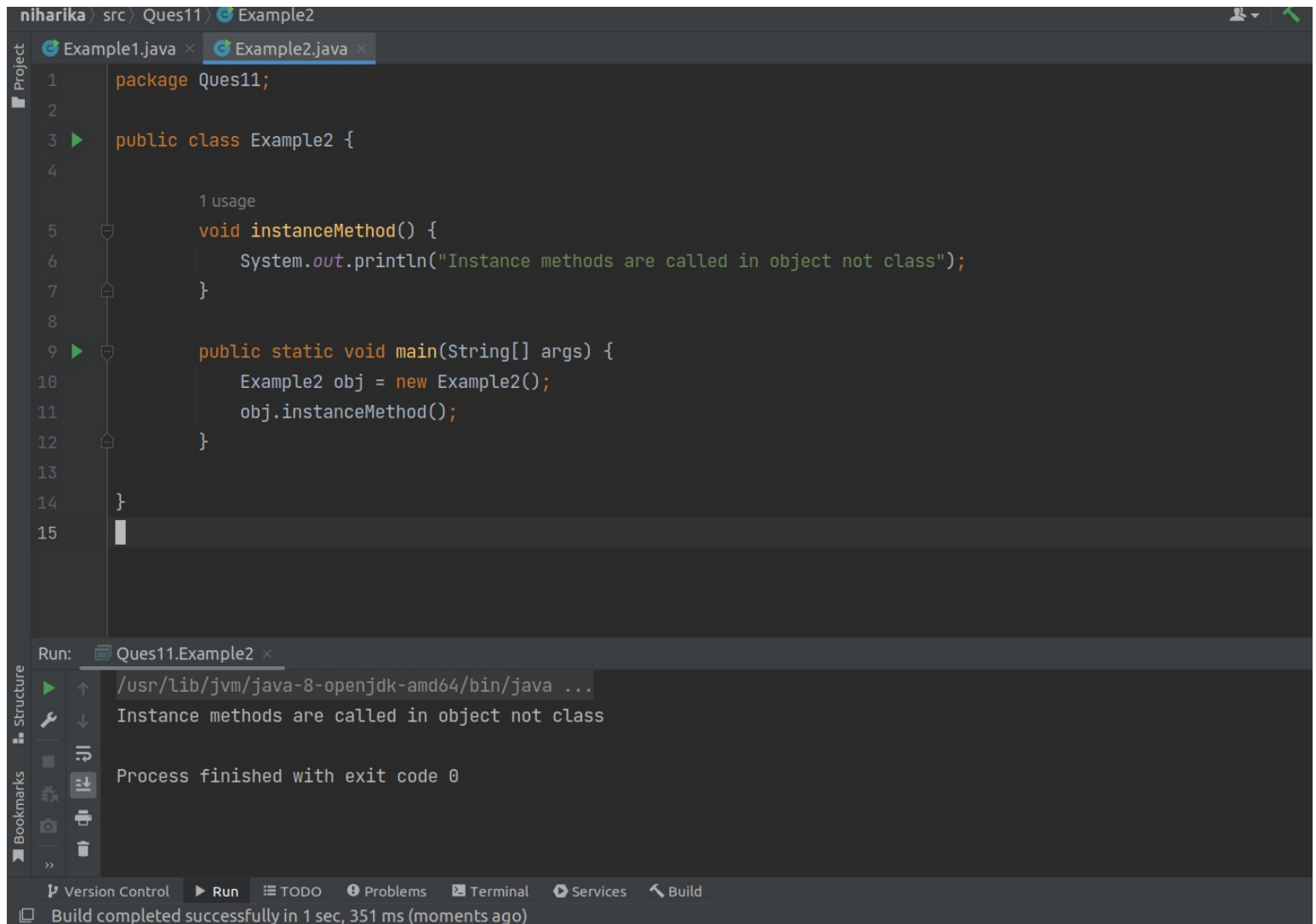
```
public class Example1 {  
  
    1 usage  
    static void staticMethod() {  
        System.out.println("This is a static method.");  
    }  
  
    1 usage  
    void instanceMethod() {  
        System.out.println("This is an instance method.");  
    }  
  
    public static void main(String[] args) {  
        Example1.staticMethod();  
  
        Example1 obj = new Example1();  
        obj.instanceMethod();  
    }  
}
```

The 'Run' output shows the execution of the program:

```
Run: Ques11.Example1 x  
/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
```

The bottom status bar indicates: 'Build completed successfully in 1 sec, 326 ms (moments ago)'.

Second Example



The screenshot shows an IDE window with the following components:

- Project Explorer:** Shows the project structure with 'Example1.java' and 'Example2.java'.
- Editor:** Displays the code for 'Example2.java'.
- Run Console:** Shows the execution output.
- Bottom Bar:** Includes tabs for Version Control, Run, TODO, Problems, Terminal, Services, and Build.

```
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();
    }
}
```

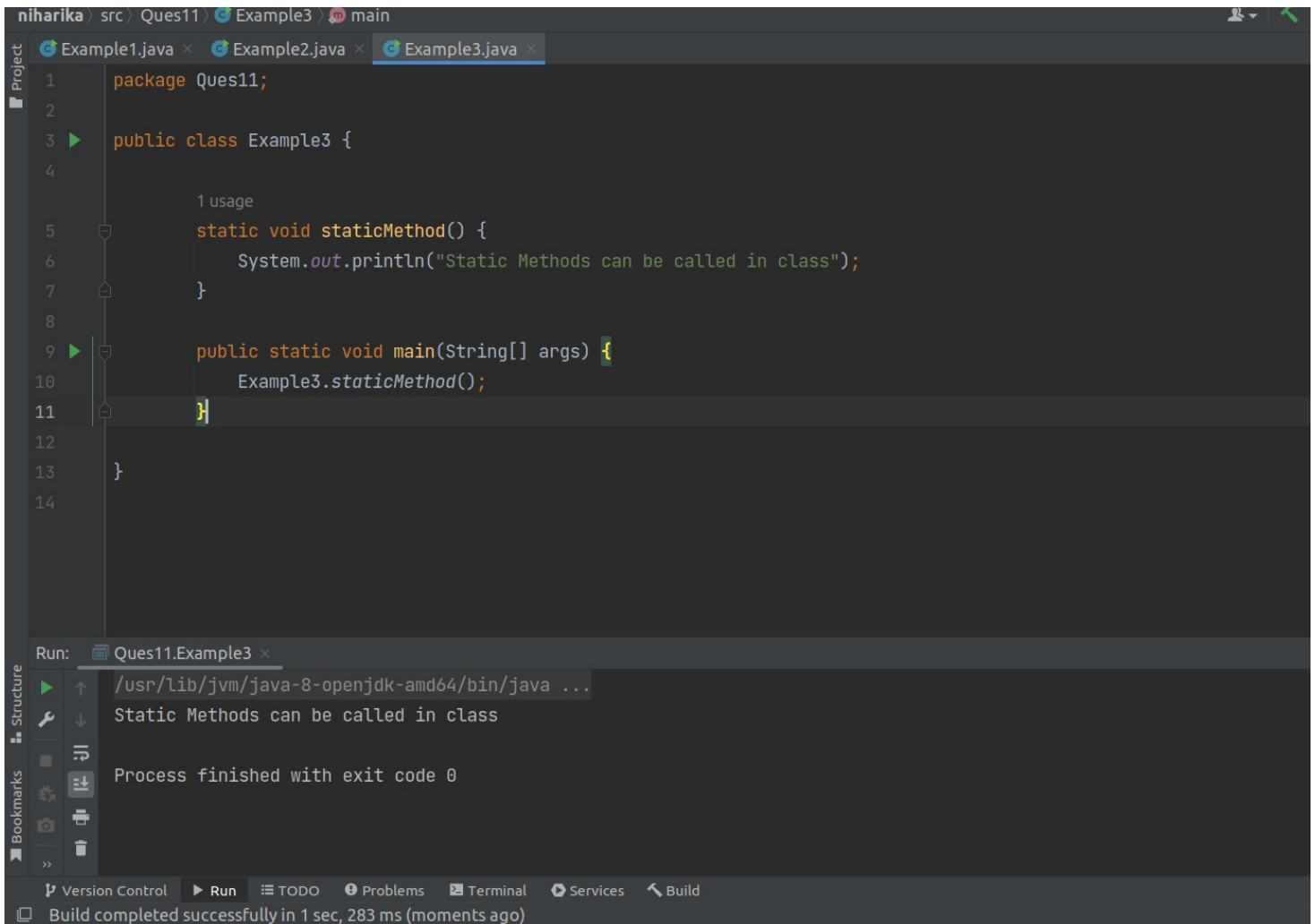
Run: Ques11.Example2 ×

```
/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:



The screenshot displays an IDE interface with a project named 'Ques11'. The 'src' folder contains three files: 'Example1.java', 'Example2.java', and 'Example3.java'. The 'Example3.java' file is open and shows the following code:

```
1 package Ques11;
2
3 public class Example3 {
4
5     1 usage
6     static void staticMethod() {
7         System.out.println("Static Methods can be called in class");
8     }
9
10    public static void main(String[] args) {
11        Example3.staticMethod();
12    }
13
14 }
```

Below the code editor, the 'Run' tab is active, showing the execution of 'Ques11.Example3'. The command executed is '/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...'. The output is 'Static Methods can be called in class'. The process finished with exit code 0.

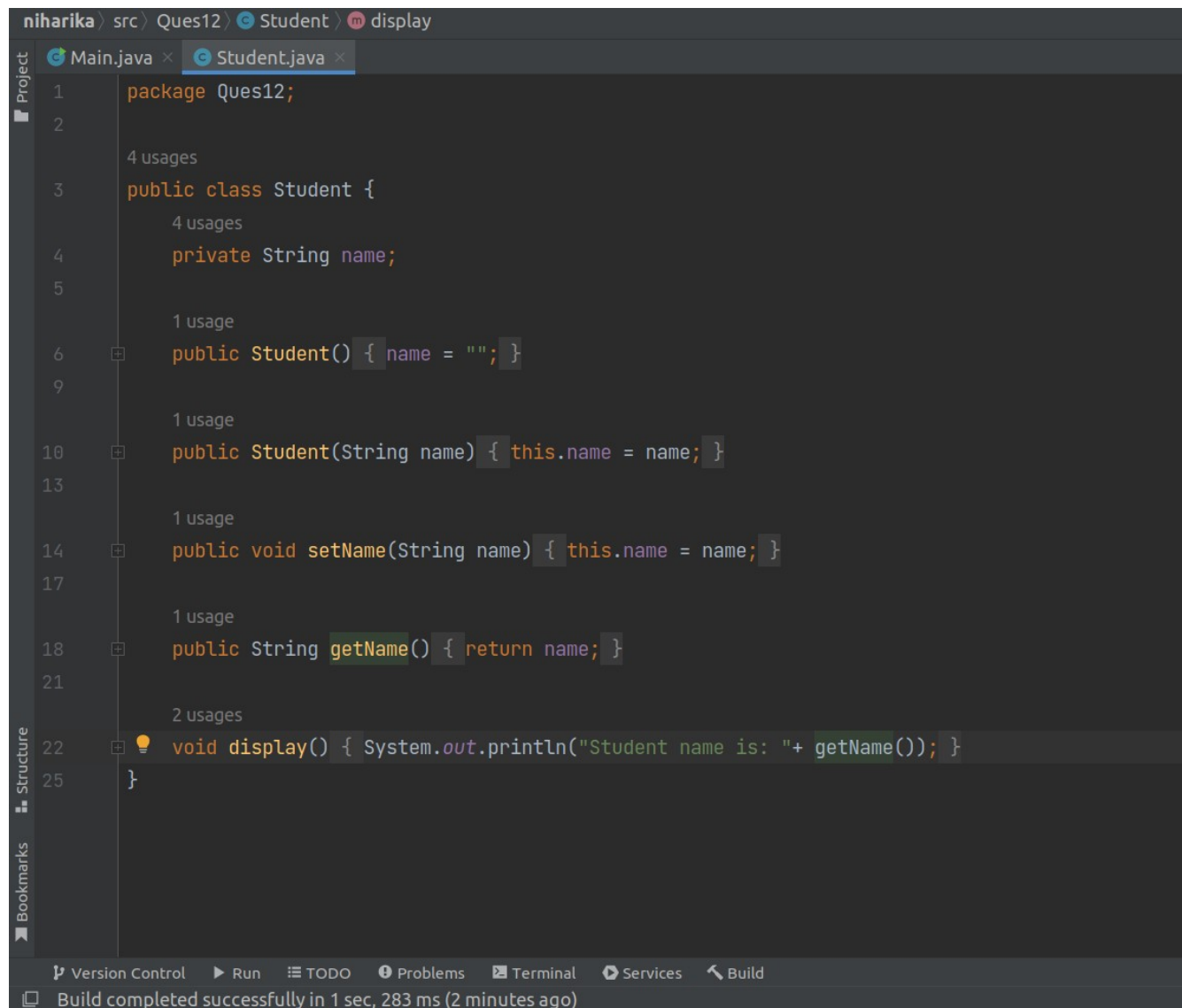
The bottom status bar indicates 'Build completed successfully in 1 sec, 283 ms (moments ago)'.

Ques no 12: Write a program of Class Student which has:

- private data member (name)
- Setter method (setName)
- Getter method (getName)

Ans:

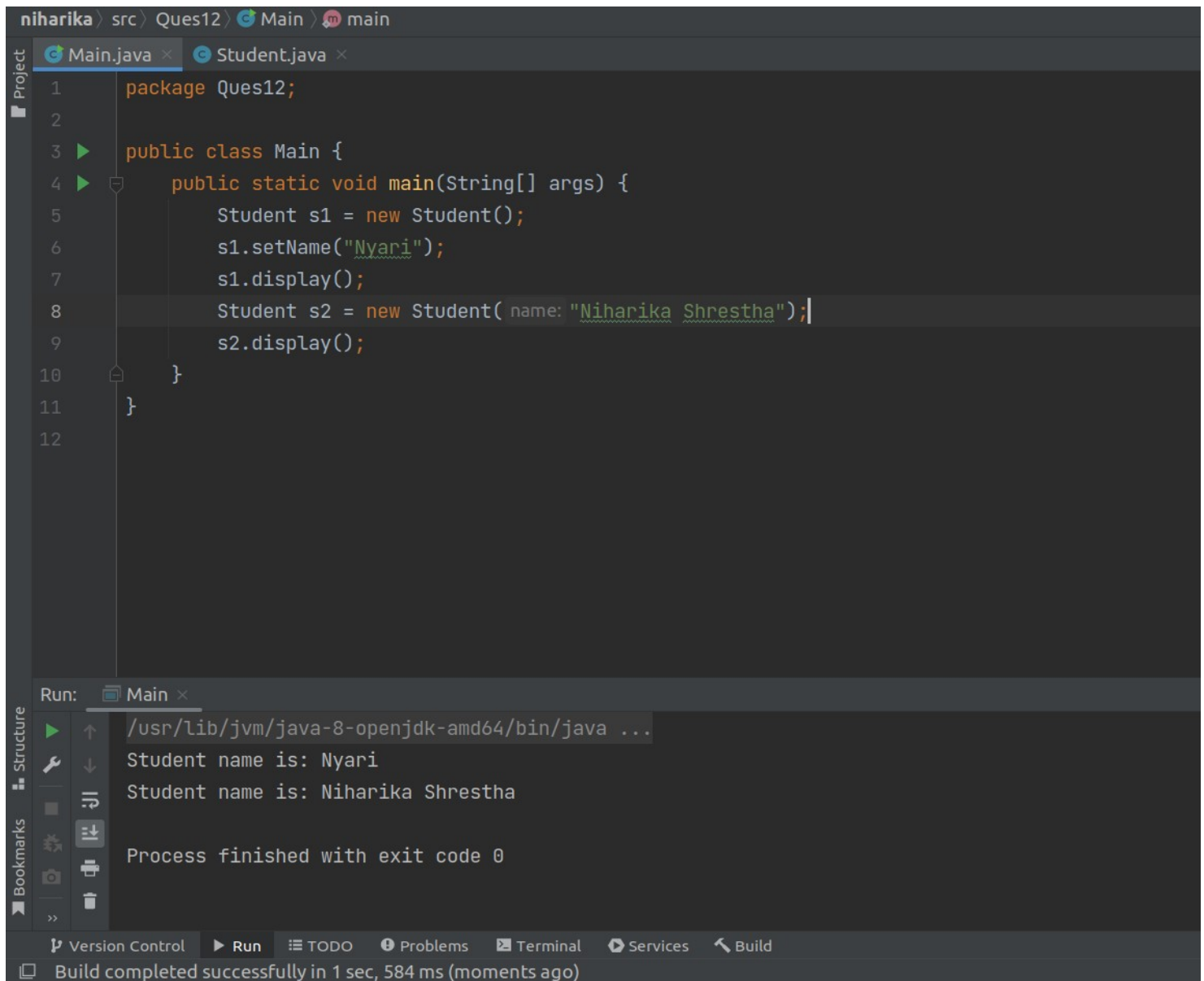
Student Class File



```
niharika > src > Ques12 > Student > display
Main.java x Student.java x
1 package Ques12;
2
3 public class Student {
4     private String name;
5
6     public Student() { name = ""; }
7
8     public Student(String name) { this.name = name; }
9
10    public void setName(String name) { this.name = name; }
11
12    public String getName() { return name; }
13
14    void display() { System.out.println("Student name is: " + getName()); }
15
16 }
```

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

Main File + Output:



The screenshot shows an IDE with two tabs: `Main.java` and `Student.java`. The `Main.java` tab is active, displaying the following code:

```
1 package Ques12;
2
3 public class Main {
4     public static void main(String[] args) {
5         Student s1 = new Student();
6         s1.setName("Nyari");
7         s1.display();
8         Student s2 = new Student( name: "Niharika Shrestha");
9         s2.display();
10    }
11 }
12
```

Below the code editor, the `Run` tab is active, showing the output of the program:

```
Run: Main x
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Student name is: Nyari
Student name is: Niharika Shrestha
Process finished with exit code 0
```

The bottom status bar indicates: `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:

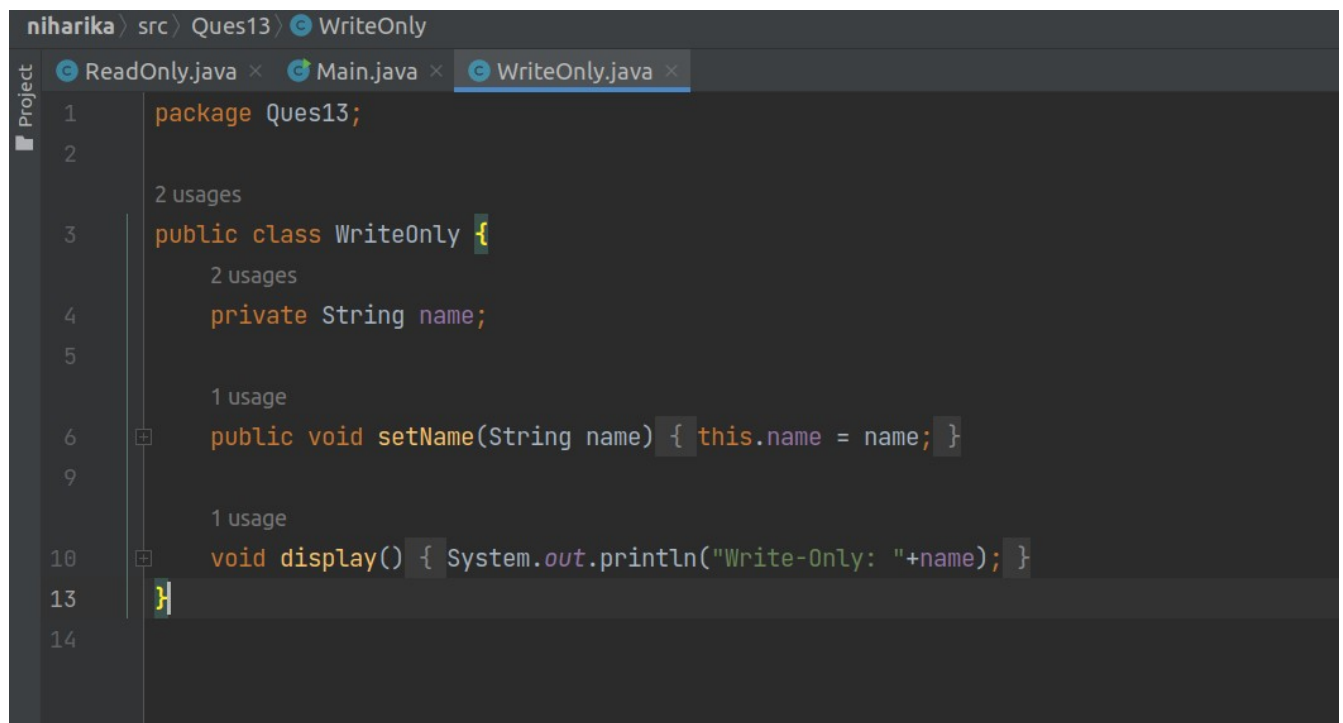
Read-Only Class File



The screenshot shows an IDE window titled "niharika > src > Ques13 > ReadOnly". The editor displays the code for the "ReadOnly.java" file. The code defines a package "Ques13" and a public class "ReadOnly". Inside the class, there is a private String member variable "name" initialized to "Nyari". There is a public String method "getName()" that returns the value of "name". There is also a void method "display()" that prints "Read-only: " followed by the value of "name".

```
1 package Ques13;
2
3 public class ReadOnly {
4     private String name = "Nyari";
5
6     public String getName() { return name; }
7
8     void display() { System.out.println("Read-only: " + name); }
9 }
10
11
12
13
14
```

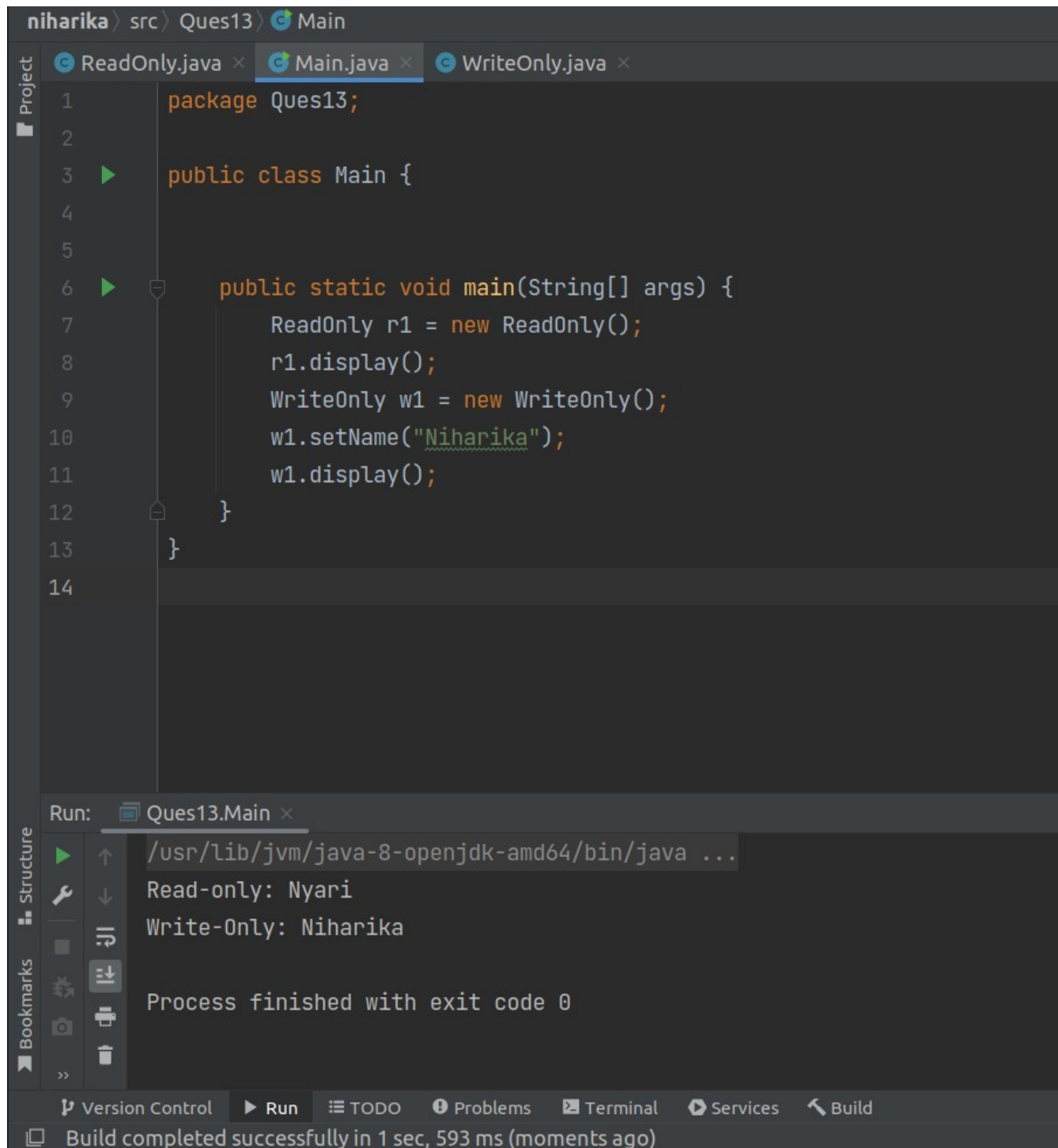
Write-Only Class File



The screenshot shows an IDE window titled "niharika > src > Ques13 > WriteOnly". The editor displays the code for the "WriteOnly.java" file. The code defines a package "Ques13" and a public class "WriteOnly". Inside the class, there is a private String member variable "name". There is a public void method "setName(String name)" that sets the value of "name" to the parameter "name". There is also a void method "display()" that prints "Write-Only: " followed by the value of "name".

```
1 package Ques13;
2
3 public class WriteOnly {
4     private String name;
5
6     public void setName(String name) { this.name = name; }
7
8     void display() { System.out.println("Write-Only: " + name); }
9 }
10
11
12
13
14
```

Main File + Output:



The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with 'Main' selected.
- Editor:** Displays the code for 'Main.java'.
- Run Console:** Shows the output of the program.
- Bottom Bar:** Contains tabs for Version Control, Run, TODO, Problems, Terminal, Services, and Build.

```
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();
    }
}
```

Run: Ques13.Main x

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

Process finished with exit code 0
```

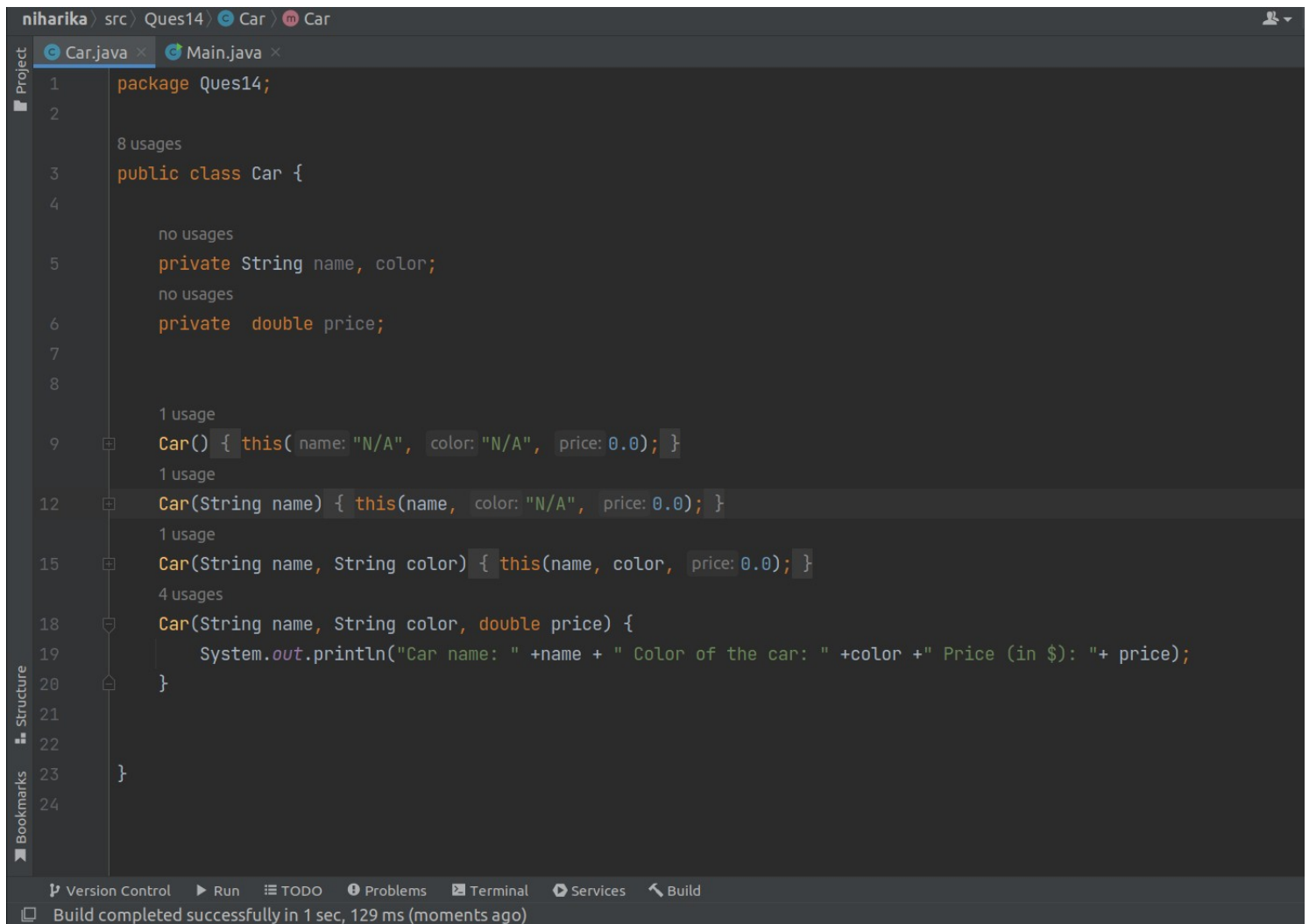
Build completed successfully in 1 sec, 593 ms (moments ago)

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:

Car Class File



```
1 package Ques14;
2
3 public class Car {
4
5     private String name, color;
6     private double price;
7
8     Car() { this(name: "N/A", color: "N/A", price: 0.0); }
9
10    Car(String name) { this(name, color: "N/A", price: 0.0); }
11
12    Car(String name, String color) { this(name, color, price: 0.0); }
13
14    Car(String name, String color, double price) {
15        System.out.println("Car name: " + name + " Color of the car: " + color + " Price (in $): " + price);
16    }
17
18 }
```

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

Main File

```
1 package Ques14;
2
3 public class Main {
4     public static void main(String[] args) {
5         Car car1 = new Car();
6         Car car2 = new Car( name: "Tesla");
7         Car car3 = new Car( name: "Rolls", color: "Black");
8         Car car4 = new Car( name: "Benzzzz", color: "Pink", price: 200000);
9     }
10 }
```

Output

```
Run: Ques14.Main x
/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: Benzzzz Color of the car: Pink Price (in $): 200000.0

Process finished with exit code 0
```

Version Control Run TODO Problems Terminal Services Build

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