

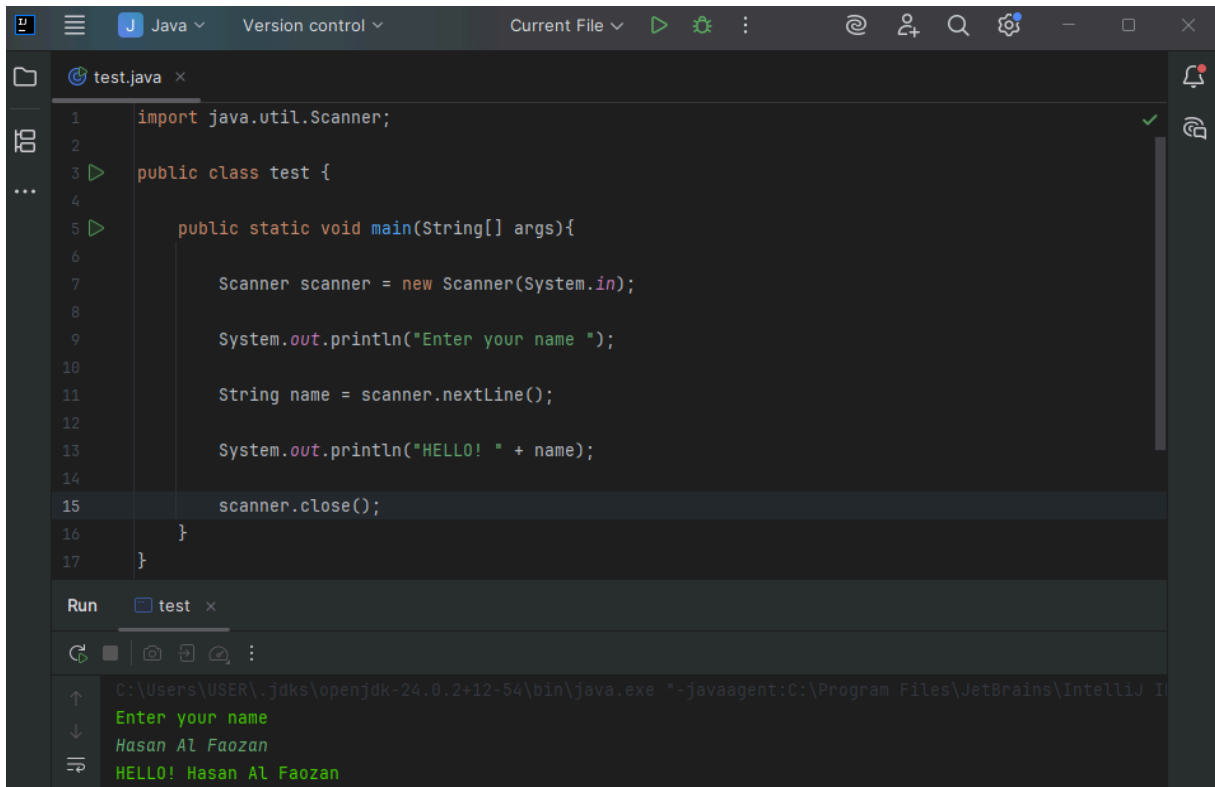
CSE 2340 - SD1 Lab Report - 1

Roll: **C243046** Section: **3BM**

- **The Steps I followed to Setup Java IDE on my pc:**

- 1) Installing Java Development Kit (JDK).
- 2) Downloading and Installing the IDE (intellij idea community edition).
- 3) Configuring the IDE:
 - a) Opening IntelliJ.
 - b) Going to File > Project Structure > SDKs.
 - c) Clicking + and add the JDK path.
 - d) Under Project, setting the default Project SDK and language level.

- **Running a program that takes my name as input and prints it on the console.**



The screenshot displays the IntelliJ IDEA interface. The main editor window shows a file named `test.java` with the following code:

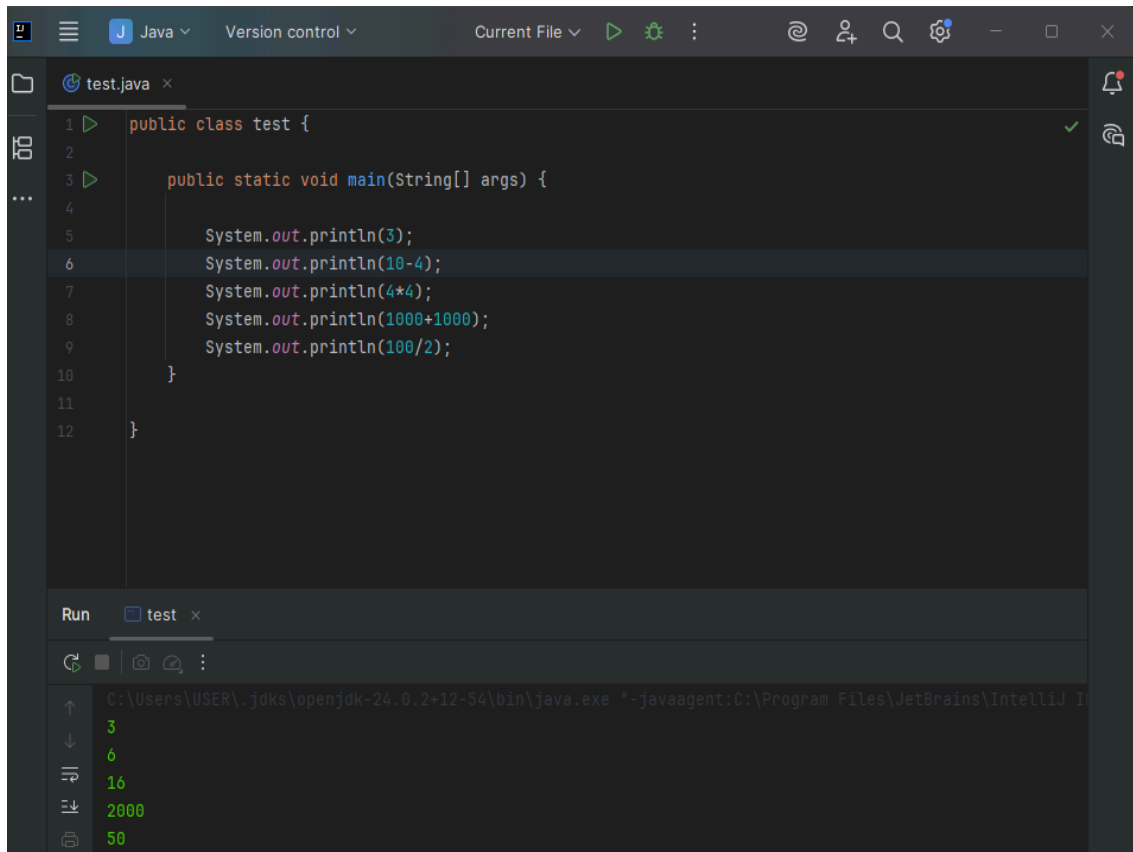
```
1 import java.util.Scanner;
2
3 public class test {
4
5     public static void main(String[] args){
6
7         Scanner scanner = new Scanner(System.in);
8
9         System.out.println("Enter your name ");
10
11         String name = scanner.nextLine();
12
13         System.out.println("HELLO! " + name);
14
15         scanner.close();
16     }
17 }
```

Below the editor, the 'Run' tab is active, showing the execution of the program. The command line at the bottom indicates the Java runtime environment: `C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I`. The console output shows the program's execution:

```
Enter your name
Hasan Al Faozan
HELLO! Hasan Al Faozan
```

- Running at least one program for each of the topics listed from the W3Schools blog.

1. Java output/print:



```
1 public class test {
2
3     public static void main(String[] args) {
4
5         System.out.println(3);
6         System.out.println(10-4);
7         System.out.println(4*4);
8         System.out.println(1000+1000);
9         System.out.println(100/2);
10    }
11
12 }
```

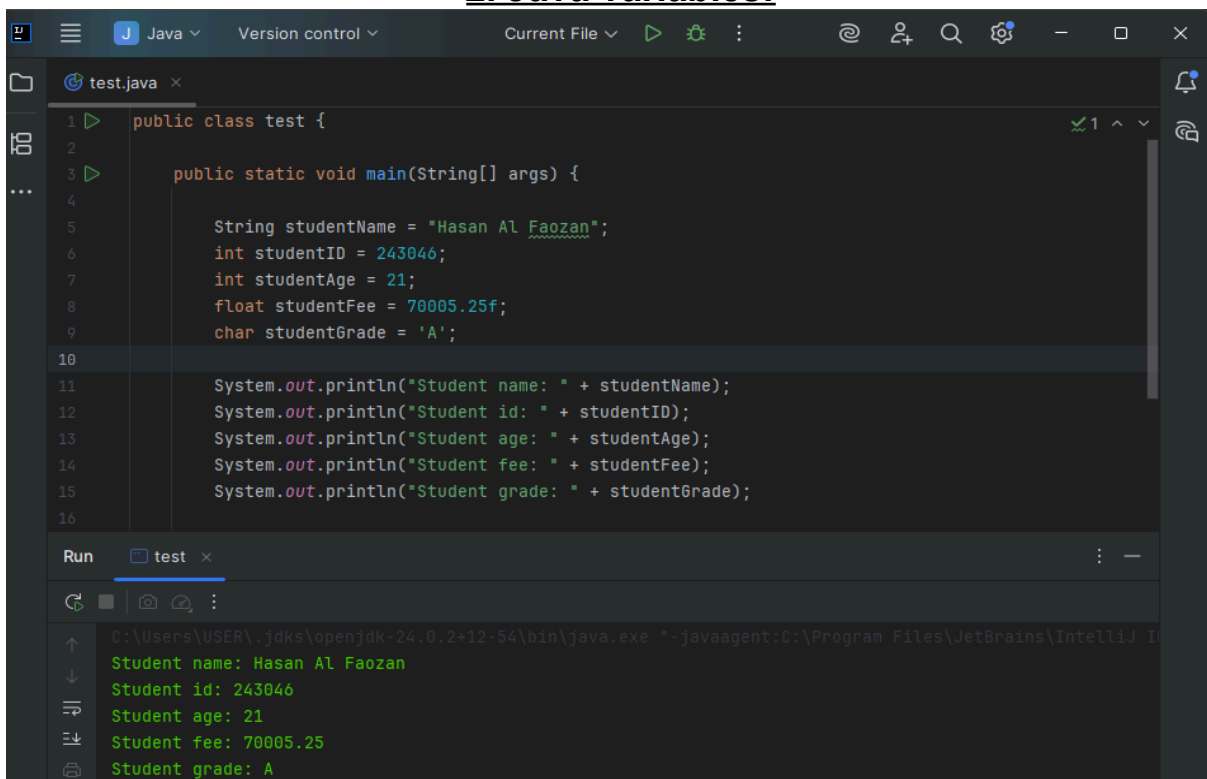
Run test x

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I

3
6
16
2000
50

☒ *This code just gives output.*

2. Java variables:



```
1 public class test {
2
3     public static void main(String[] args) {
4
5         String studentName = "Hasan Al Faozan";
6         int studentID = 243046;
7         int studentAge = 21;
8         float studentFee = 70005.25f;
9         char studentGrade = 'A';
10
11         System.out.println("Student name: " + studentName);
12         System.out.println("Student id: " + studentID);
13         System.out.println("Student age: " + studentAge);
14         System.out.println("Student fee: " + studentFee);
15         System.out.println("Student grade: " + studentGrade);
16    }
17 }
```

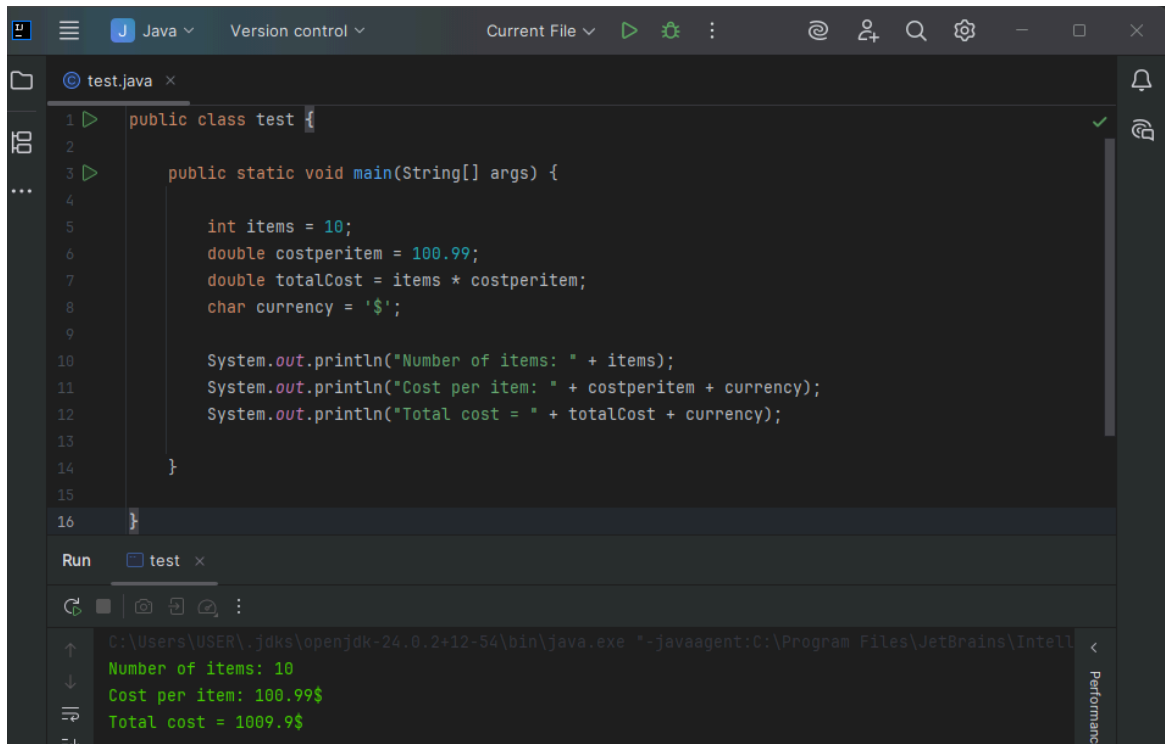
Run test x

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I

Student name: Hasan Al Faozan
Student id: 243046
Student age: 21
Student fee: 70005.25
Student grade: A

☑ ***This code shows basic Java variables use.***

3. Java Data Types:



The screenshot shows the IntelliJ IDEA IDE with a Java file named `test.java`. The code defines a public class `test` with a `main` method. Inside `main`, four variables are declared and initialized: `int items = 10;`, `double costperitem = 100.99;`, `double totalCost = items * costperitem;`, and `char currency = '$';`. These are followed by three `System.out.println` statements that output the values of these variables. The Run window at the bottom shows the execution output: `Number of items: 10`, `Cost per item: 100.99$`, and `Total cost = 1009.9$`.

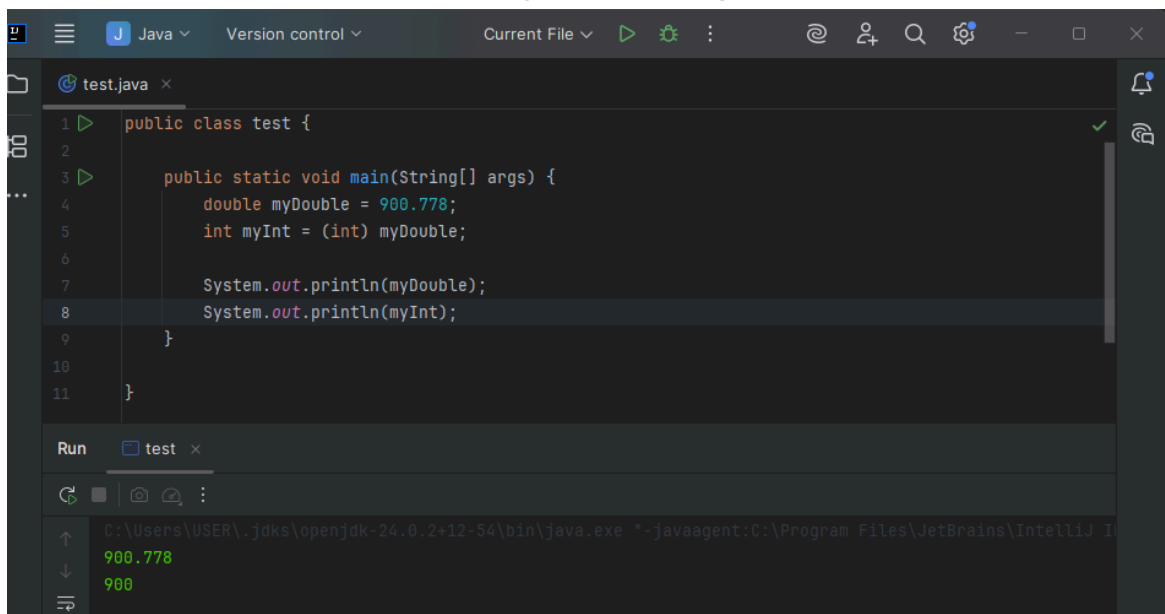
```
1 public class test {
2
3     public static void main(String[] args) {
4
5         int items = 10;
6         double costperitem = 100.99;
7         double totalCost = items * costperitem;
8         char currency = '$';
9
10        System.out.println("Number of items: " + items);
11        System.out.println("Cost per item: " + costperitem + currency);
12        System.out.println("Total cost = " + totalCost + currency);
13    }
14 }
15
16 }
```

Run test x

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I
Number of items: 10
Cost per item: 100.99\$
Total cost = 1009.9\$

☑ ***This code shows Java Data Types.***

4. Java Type Casting:



The screenshot shows the IntelliJ IDEA IDE with a Java file named `test.java`. The code defines a public class `test` with a `main` method. Inside `main`, a `double` variable `myDouble` is initialized with the value `900.778`, and an `int` variable `myInt` is initialized with the value `(int) myDouble`, demonstrating explicit type casting. These are followed by two `System.out.println` statements that output the values of `myDouble` and `myInt`. The Run window at the bottom shows the execution output: `900.778` and `900`.

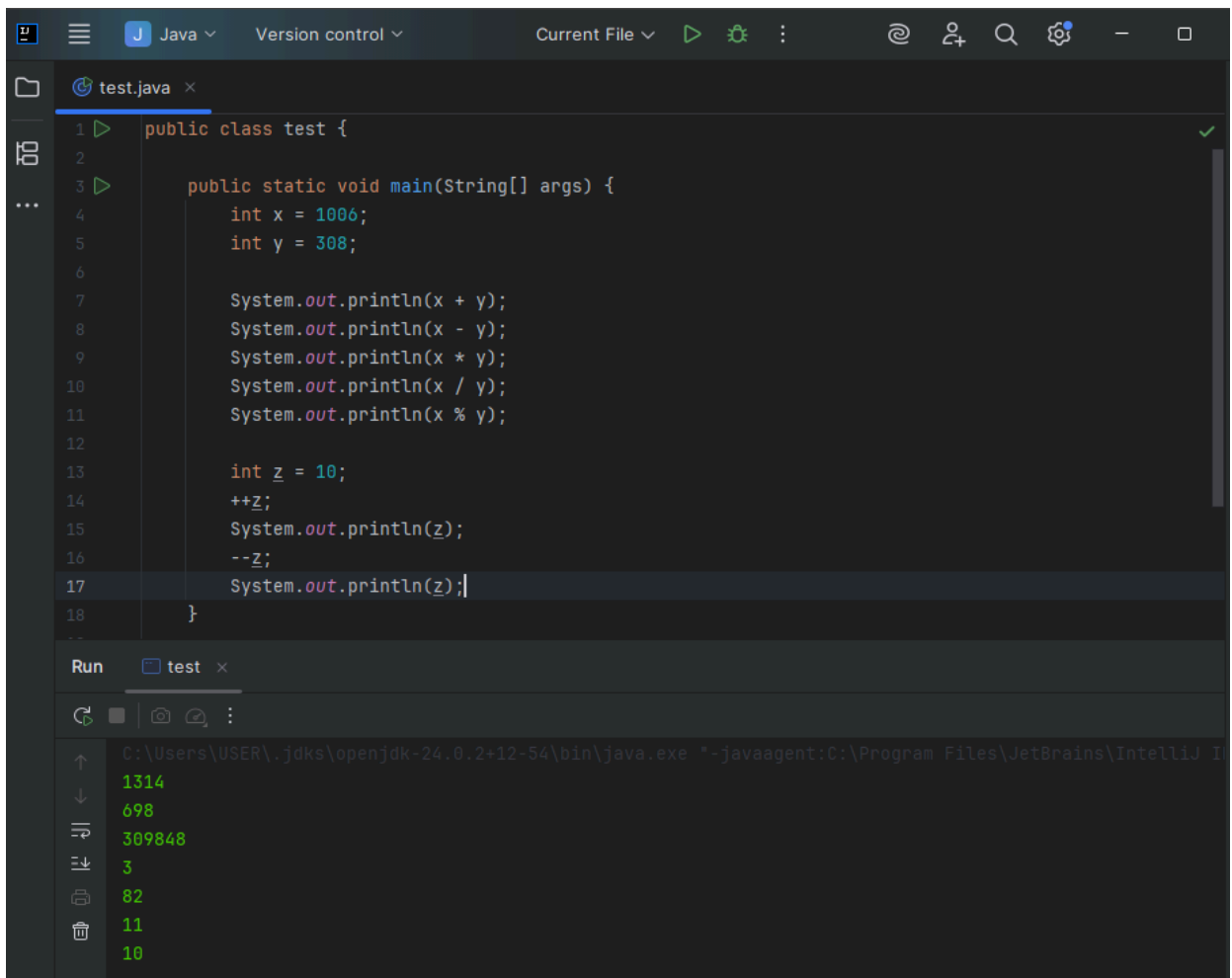
```
1 public class test {
2
3     public static void main(String[] args) {
4         double myDouble = 900.778;
5         int myInt = (int) myDouble;
6
7         System.out.println(myDouble);
8         System.out.println(myInt);
9     }
10 }
11 }
```

Run test x

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I
900.778
900

☑ ***This code shows basic Java Type Casting where we can change a data type to another data type***

5. Java Operators:



```
1 public class test {
2
3     public static void main(String[] args) {
4         int x = 1006;
5         int y = 308;
6
7         System.out.println(x + y);
8         System.out.println(x - y);
9         System.out.println(x * y);
10        System.out.println(x / y);
11        System.out.println(x % y);
12
13        int z = 10;
14        ++z;
15        System.out.println(z);
16        --z;
17        System.out.println(z);
18    }
19 }
```

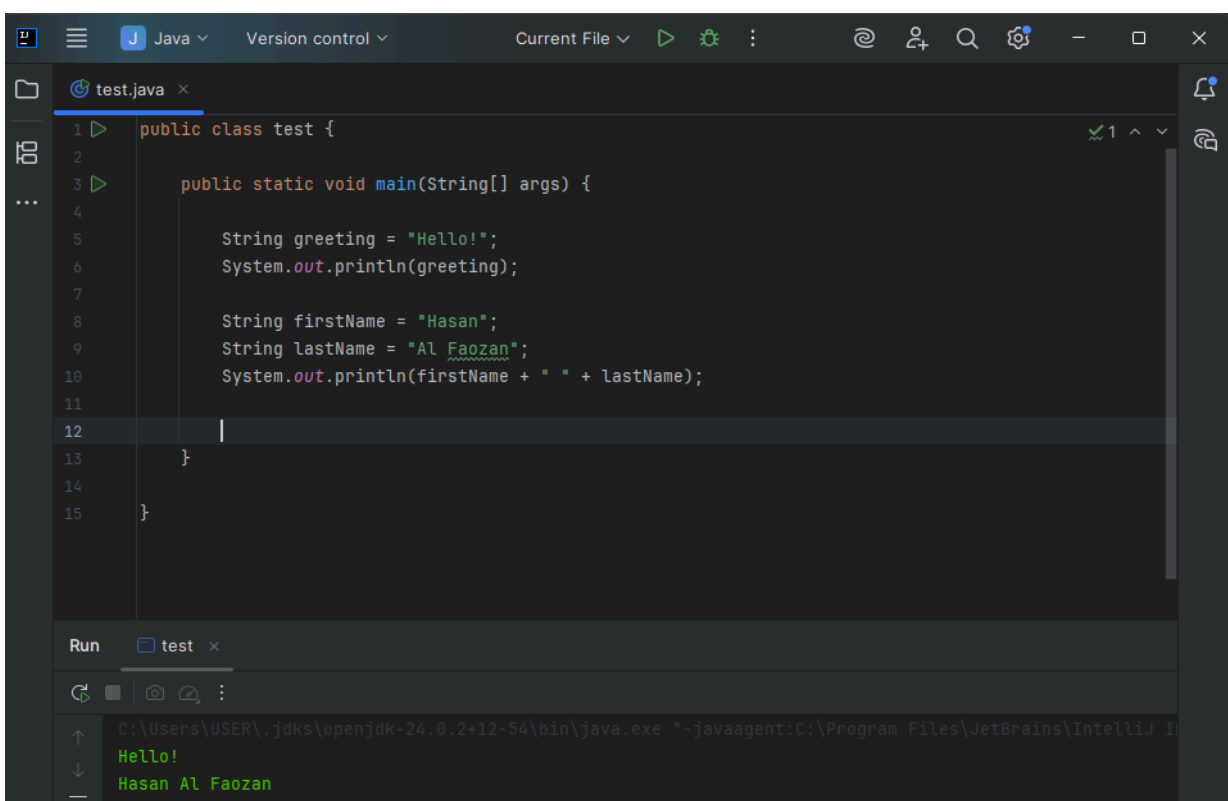
Run test

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I

1314
698
309848
3
82
11
10

✓ ***This simple code shows all the Java Operators and their use. .***

6. Java String:



```
1 public class test {
2
3     public static void main(String[] args) {
4
5         String greeting = "Hello!";
6         System.out.println(greeting);
7
8         String firstName = "Hasan";
9         String lastName = "Al Faozan";
10        System.out.println(firstName + " " + lastName);
11
12    }
13
14 }
15 }
```

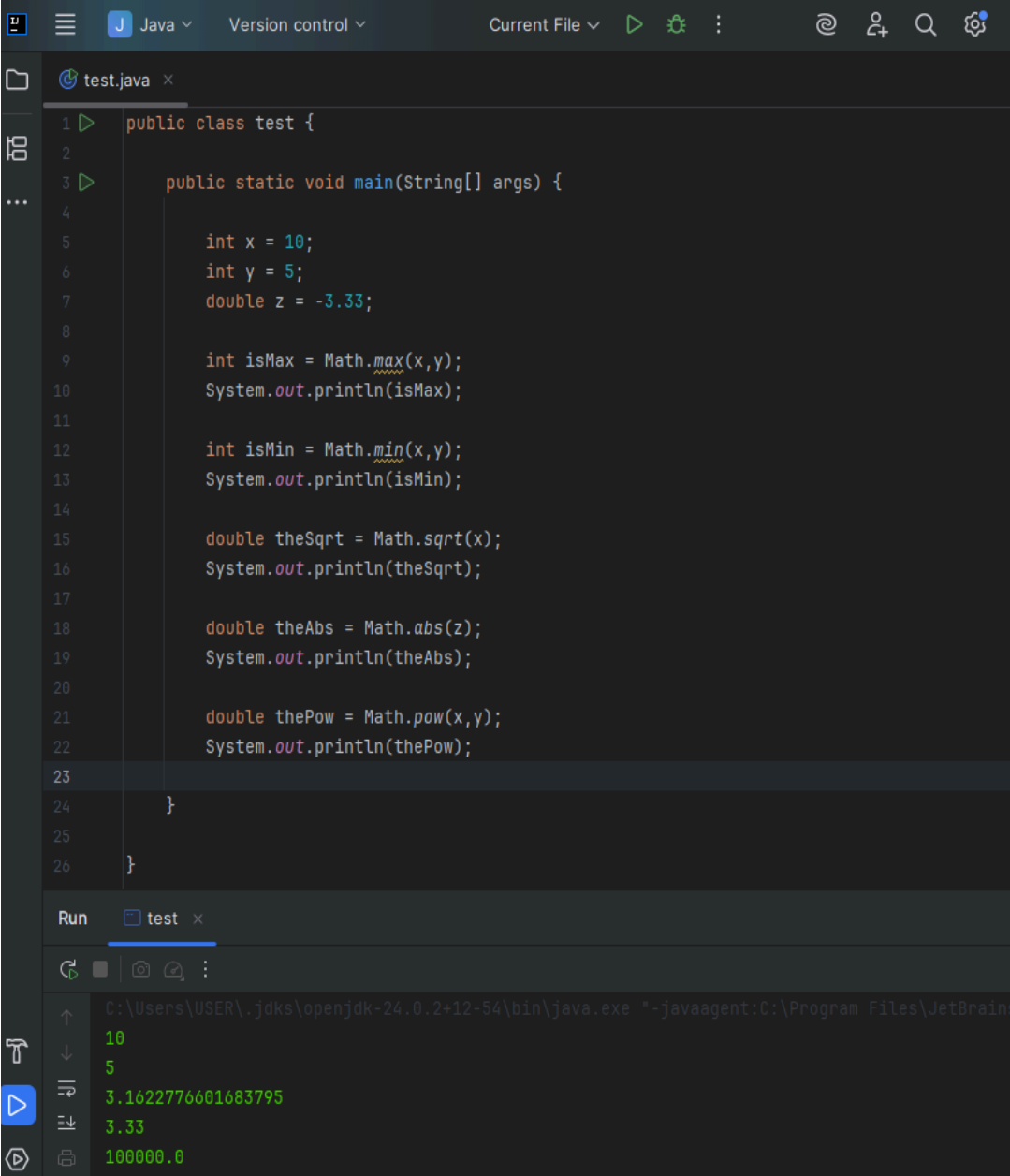
Run test

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I

Hello!
Hasan Al Faozan

☒ ***This code gives String output.***

7. Java Math:



The screenshot shows an IDE window with a Java file named 'test.java'. The code defines a public class 'test' with a static method 'main'. Inside 'main', variables 'x' (10), 'y' (5), and 'z' (-3.33) are declared. The code then uses the 'Math' class to calculate 'isMax' (Math.max), 'isMin' (Math.min), 'theSqrt' (Math.sqrt), 'theAbs' (Math.abs), and 'thePow' (Math.pow), printing each result. Below the code editor, the 'Run' tab is active, showing the command used to execute the program and the resulting output: 10, 5, 3.1622776601683795, 3.33, and 100000.0.

```
1 public class test {  
2  
3     public static void main(String[] args) {  
4  
5         int x = 10;  
6         int y = 5;  
7         double z = -3.33;  
8  
9         int isMax = Math.max(x,y);  
10        System.out.println(isMax);  
11  
12        int isMin = Math.min(x,y);  
13        System.out.println(isMin);  
14  
15        double theSqrt = Math.sqrt(x);  
16        System.out.println(theSqrt);  
17  
18        double theAbs = Math.abs(z);  
19        System.out.println(theAbs);  
20  
21        double thePow = Math.pow(x,y);  
22        System.out.println(thePow);  
23  
24    }  
25  
26 }
```

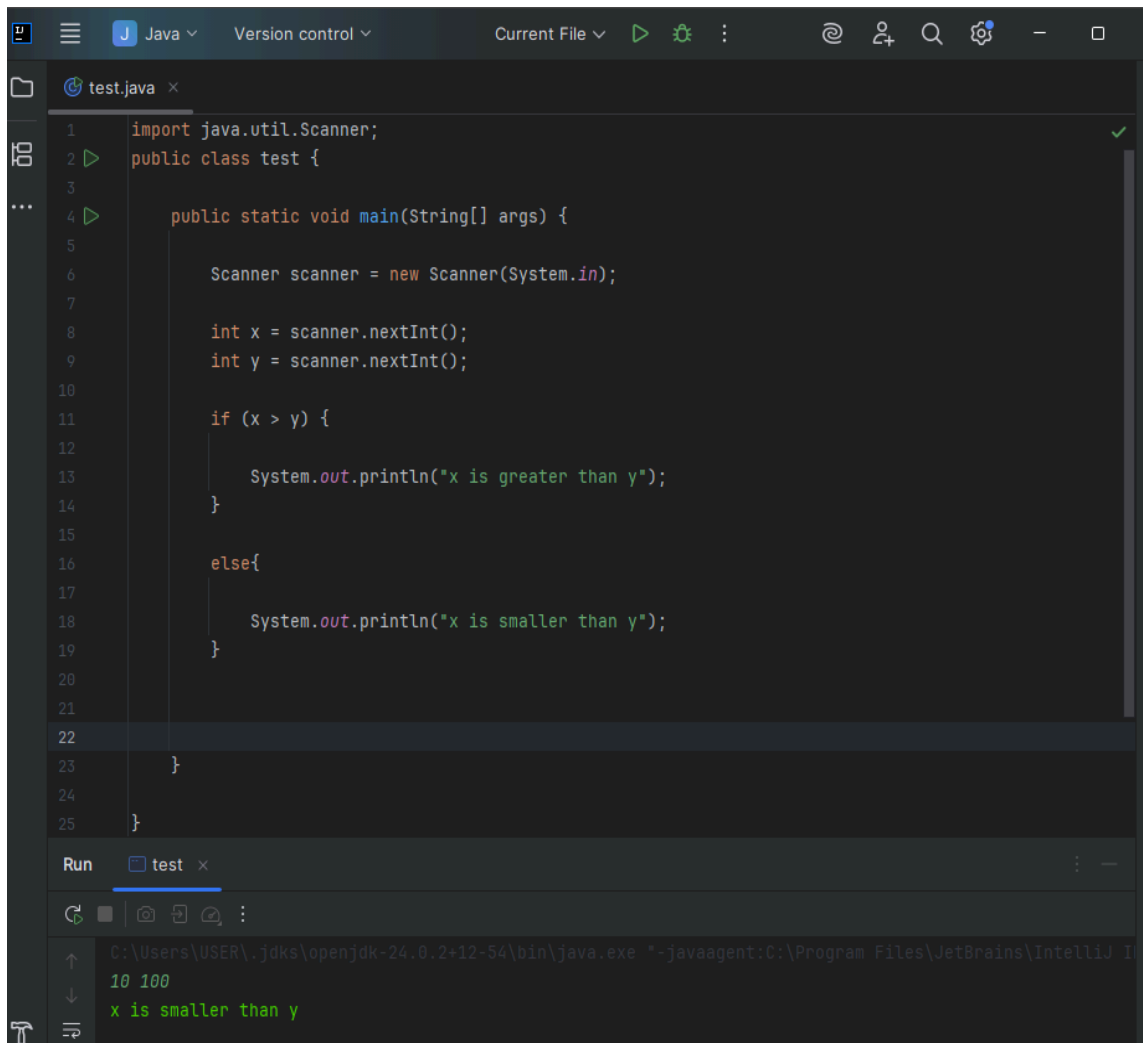
Run test x

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrain

10
5
3.1622776601683795
3.33
100000.0

☒ ***This code shows all the math functions and their use in java.***

8. Java If...Else:



The screenshot shows the IntelliJ IDEA IDE with a Java file named `test.java`. The code implements an `if...else` statement to compare two integers, `x` and `y`, entered by the user. The code is as follows:

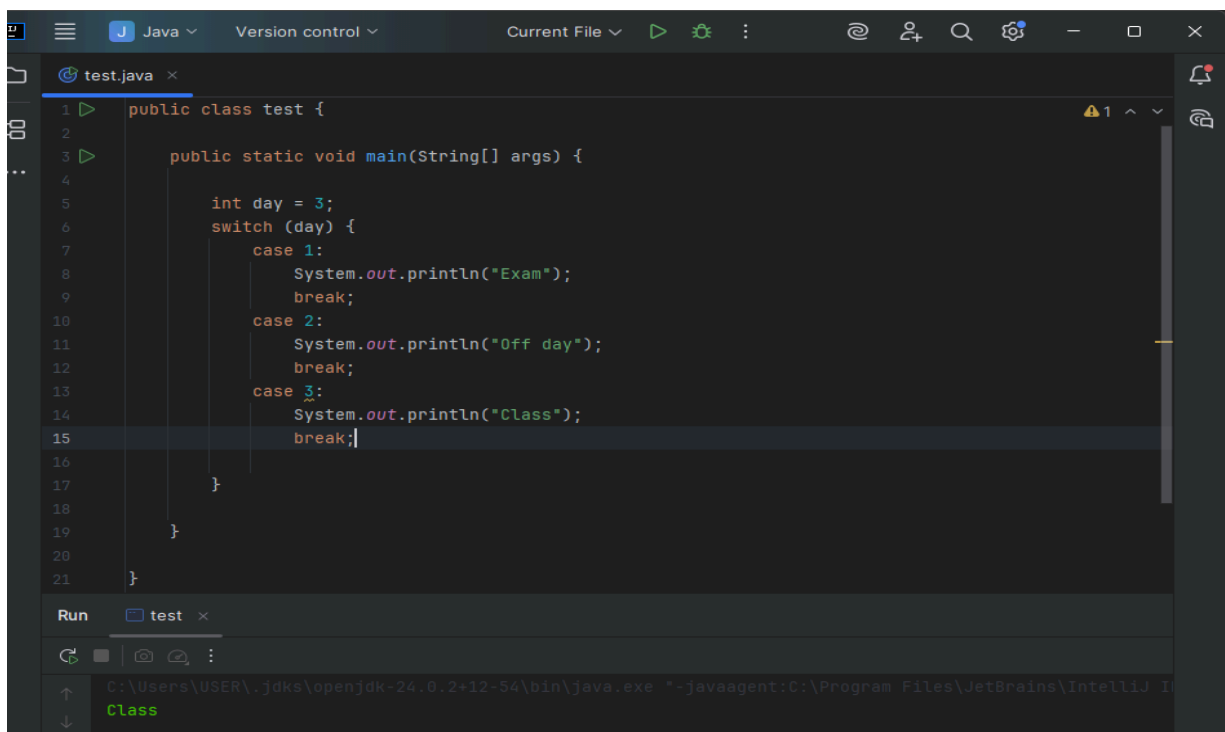
```
1 import java.util.Scanner;
2 public class test {
3
4     public static void main(String[] args) {
5
6         Scanner scanner = new Scanner(System.in);
7
8         int x = scanner.nextInt();
9         int y = scanner.nextInt();
10
11         if (x > y) {
12             System.out.println("x is greater than y");
13         }
14
15         else{
16             System.out.println("x is smaller than y");
17         }
18     }
19 }
20
21
22
23
24
25
```

Below the code editor, the Run tab shows the execution output:

```
Run test x
C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ I
10 100
x is smaller than y
```

☒ ***This Java code compares two integers using if...else.***

9. Java Switch:



The screenshot shows the IntelliJ IDEA IDE with a Java file named `test.java`. The code implements a `switch` statement to print different messages based on the value of the variable `day`. The code is as follows:

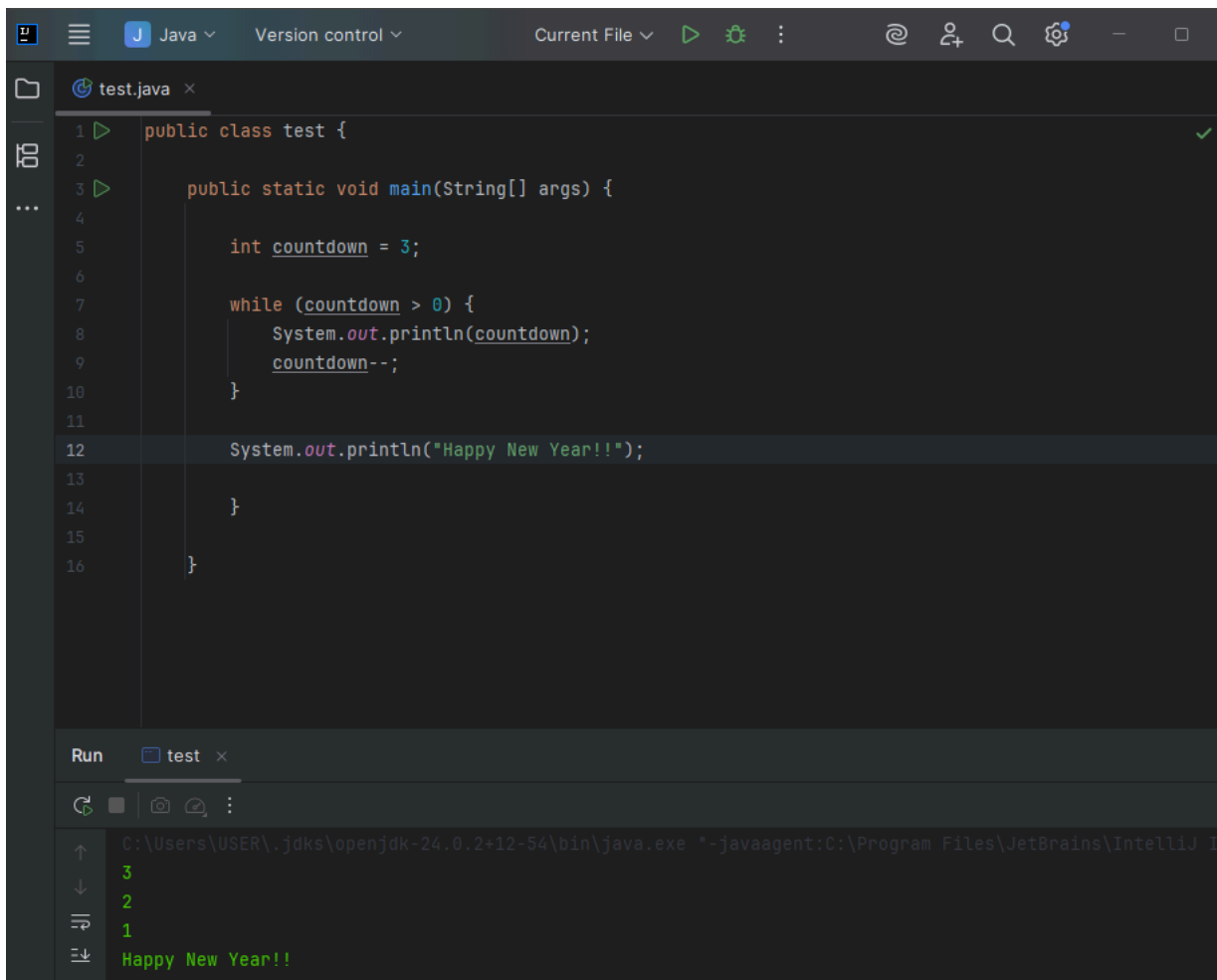
```
1 public class test {
2
3     public static void main(String[] args) {
4
5         int day = 3;
6         switch (day) {
7             case 1:
8                 System.out.println("Exam");
9                 break;
10            case 2:
11                System.out.println("Off day");
12                break;
13            case 3:
14                System.out.println("Class");
15                break;
16        }
17    }
18 }
19
20
21
```

Below the code editor, the Run tab shows the execution output:

```
Run test x
C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ I
Class
```

☒ ***This code shows the basic use of Switch functions in Java***

10. Java While Loop.



The screenshot shows the IntelliJ IDEA IDE with a Java file named `test.java`. The code defines a `public class test` with a `main` method. Inside the `main` method, a variable `countdown` is initialized to 3. A `while` loop is used to print the value of `countdown` as long as it is greater than 0, and then decrement it. After the loop, a message "Happy New Year!!" is printed. The Run console shows the output: 3, 2, 1, and Happy New Year!!.

```
1 public class test {
2
3     public static void main(String[] args) {
4
5         int countdown = 3;
6
7         while (countdown > 0) {
8             System.out.println(countdown);
9             countdown--;
10        }
11
12        System.out.println("Happy New Year!!");
13
14    }
15
16 }
```

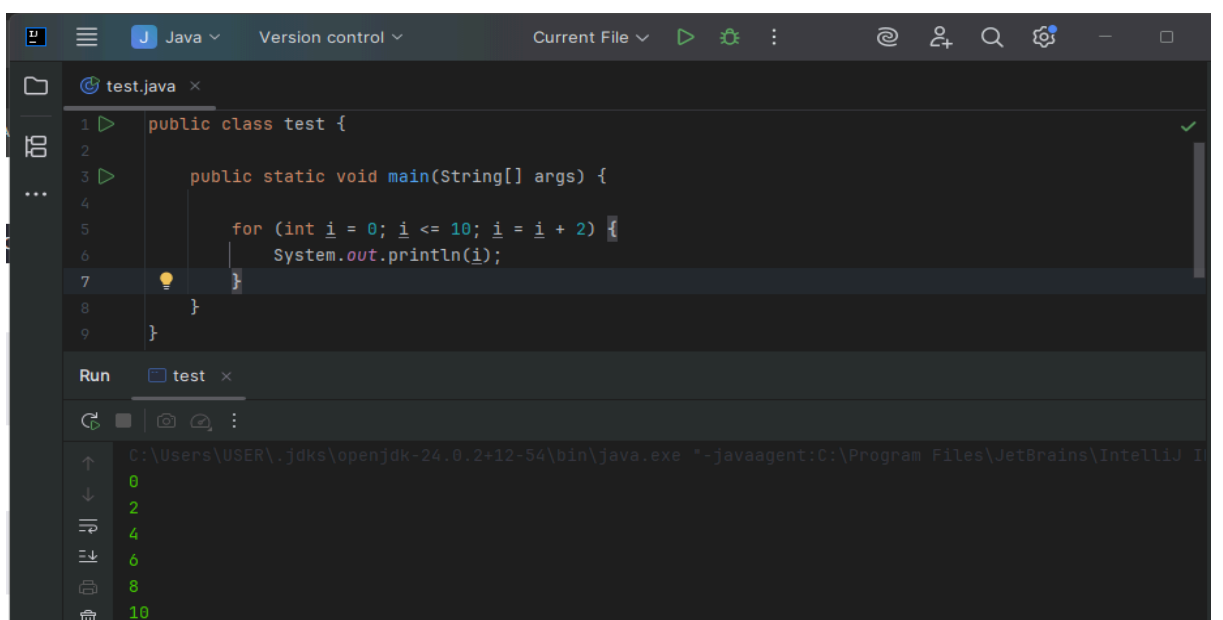
Run test x

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I

3
2
1
Happy New Year!!

☒ ***This code shows simple while loop use in Java.***

11. Java For Loop:



The screenshot shows the IntelliJ IDEA IDE with a Java file named `test.java`. The code defines a `public class test` with a `main` method. Inside the `main` method, a `for` loop is used to print the values of `i` from 0 to 10, incrementing by 2. The Run console shows the output: 0, 2, 4, 6, 8, 10.

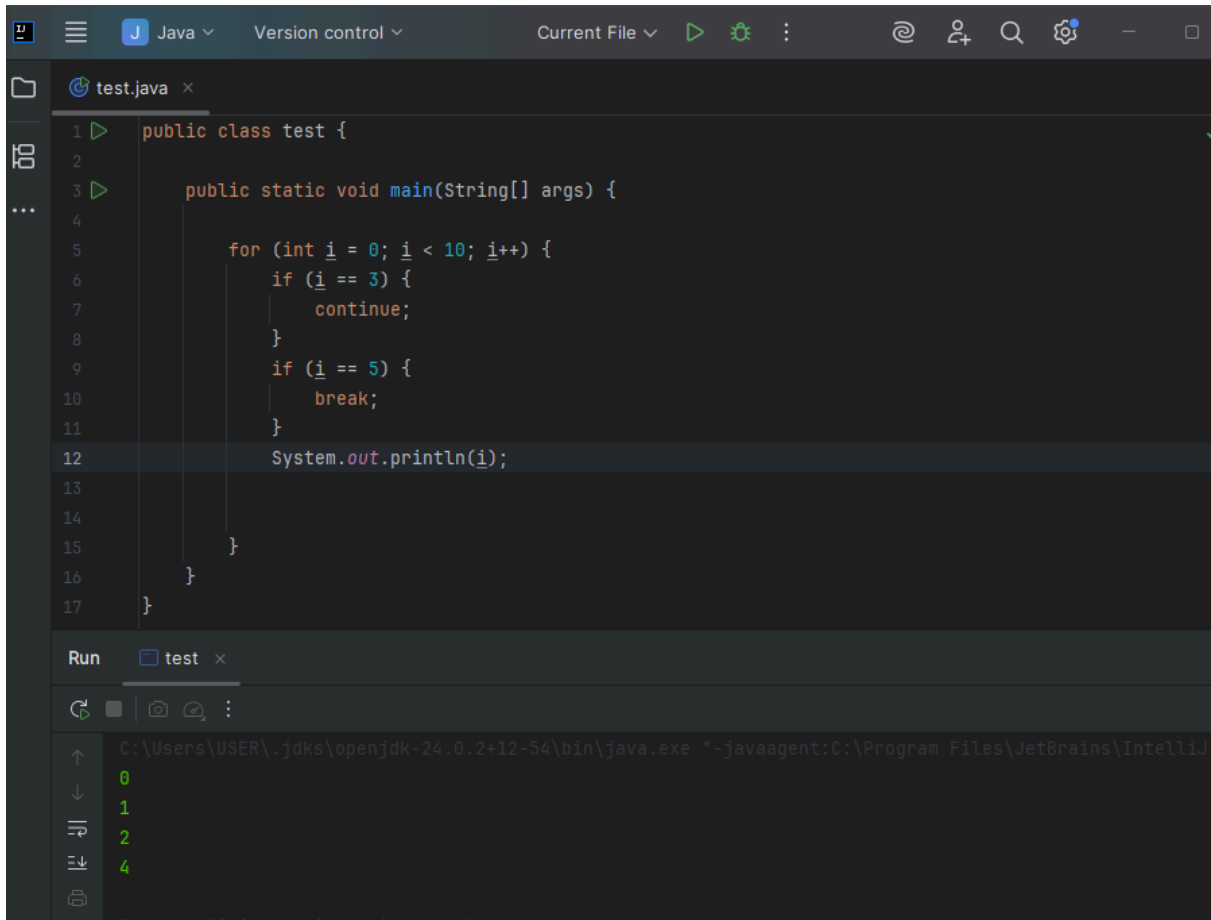
```
1 public class test {
2
3     public static void main(String[] args) {
4
5         for (int i = 0; i <= 10; i = i + 2) {
6             System.out.println(i);
7         }
8     }
9 }
```

Run test x

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I

0
2
4
6
8
10

12. Java Break and Continue:



The screenshot shows the IntelliJ IDEA IDE with a Java file named `test.java`. The code defines a `test` class with a `main` method. Inside the `main` method, there is a `for` loop that iterates from `i = 0` to `i = 10`. The loop contains two conditional statements: `if (i == 3) { continue; }` and `if (i == 5) { break; }`. After the loop, `System.out.println(i);` is executed. The Run window shows the output: `0`, `1`, `2`, and `4`, indicating that the loop was terminated at `i = 5` due to the `break` statement.

```
1 public class test {
2
3     public static void main(String[] args) {
4
5         for (int i = 0; i < 10; i++) {
6             if (i == 3) {
7                 continue;
8             }
9             if (i == 5) {
10                break;
11            }
12            System.out.println(i);
13
14        }
15    }
16 }
17 }
```

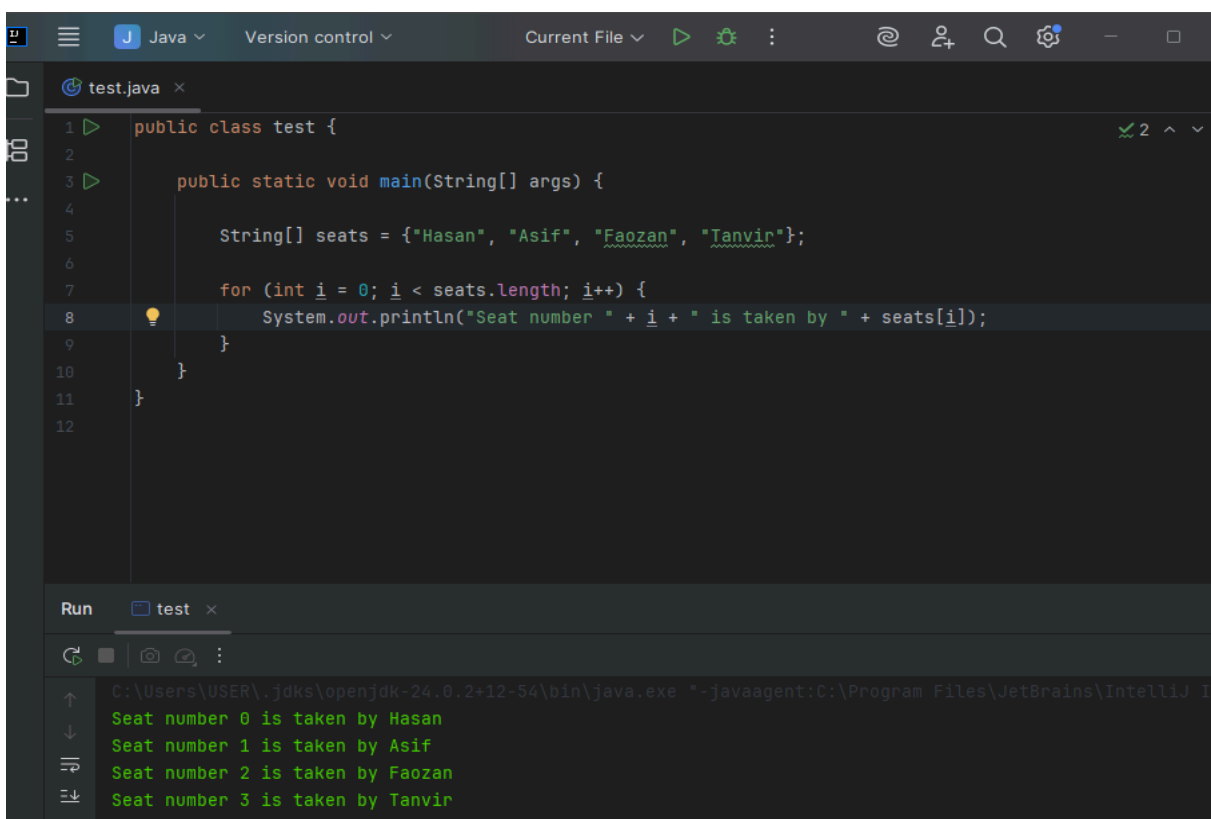
Run test x

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ

0
1
2
4

✓ ***This code shows Break and Continue statements used in Java.***

13. Java Arrays.



The screenshot shows the IntelliJ IDEA IDE with a Java file named `test.java`. The code defines a `test` class with a `main` method. Inside the `main` method, an array `seats` is declared and initialized with the values `"Hasan"`, `"Asif"`, `"Faazan"`, and `"Tanvir"`. A `for` loop iterates from `i = 0` to `i = 3`, printing the seat number and the corresponding name from the `seats` array. The Run window shows the output: `Seat number 0 is taken by Hasan`, `Seat number 1 is taken by Asif`, `Seat number 2 is taken by Faazan`, and `Seat number 3 is taken by Tanvir`.

```
1 public class test {
2
3     public static void main(String[] args) {
4
5         String[] seats = {"Hasan", "Asif", "Faazan", "Tanvir"};
6
7         for (int i = 0; i < seats.length; i++) {
8             System.out.println("Seat number " + i + " is taken by " + seats[i]);
9         }
10    }
11 }
12 }
```

Run test x

C:\Users\USER\.jdk\openjdk-24.0.2+12-54\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I

Seat number 0 is taken by Hasan
Seat number 1 is taken by Asif
Seat number 2 is taken by Faazan
Seat number 3 is taken by Tanvir

☒ ***This code shows simple use of Array in Java.***