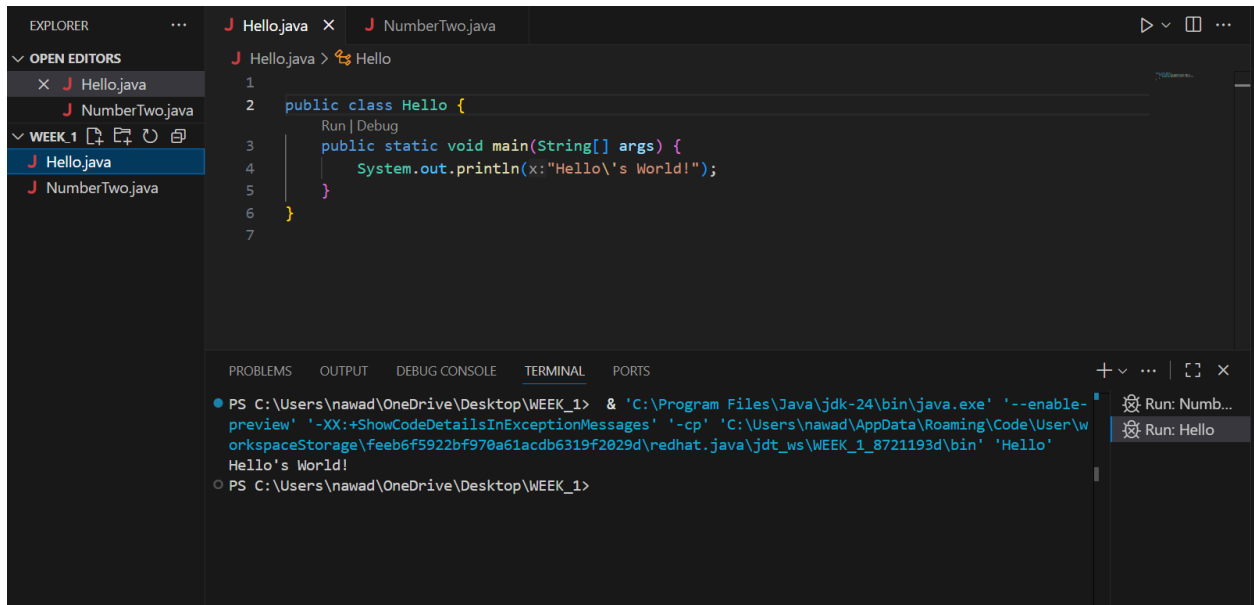


1. Write a program that prints **Hello World!** to the console.
Save the program as Hello.java.



The screenshot shows an IDE with two open files: Hello.java and NumberTwo.java. The Hello.java file contains the following code:

```
1  
2 public class Hello {  
3     |  
4     | public static void main(String[] args) {  
5         | System.out.println(x:"Hello's World!");  
6     | }  
7 }
```

The terminal output shows the command executed and the resulting output:

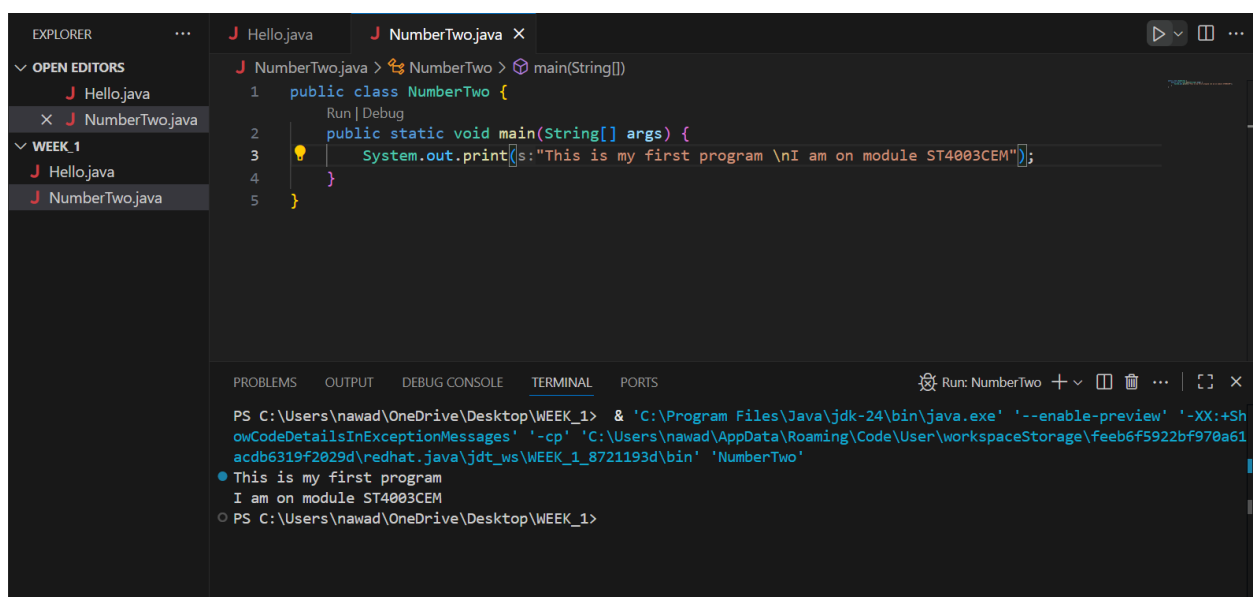
```
PS C:\Users\nawad\OneDrive\Desktop\WEEK_1> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-  
preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\nawad\AppData\Roaming\Code\User\w  
orkspaceStorage\feeb6f5922bf970a61acdb6319f2029d\redhat.java\jdt_ws\WEEK_1_8721193d\bin' 'Hello'  
Hello's World!  
PS C:\Users\nawad\OneDrive\Desktop\WEEK_1>
```

2. Extend/Modify the above program to print two additional lines:

This is my first program

I am on module ST4003CEM

System.out.print (**not** println) and produces the same output.



The screenshot shows an IDE with two open files: Hello.java and NumberTwo.java. The NumberTwo.java file contains the following code:

```
1 public class NumberTwo {  
2     |  
3     | public static void main(String[] args) {  
4         | System.out.print(s:"This is my first program \nI am on module ST4003CEM");  
5     | }  
6 }
```

The terminal output shows the command executed and the resulting output:

```
PS C:\Users\nawad\OneDrive\Desktop\WEEK_1> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-  
preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\nawad\AppData\Roaming\Code\User\w  
orkspaceStorage\feeb6f5922bf970a61acdb6319f2029d\redhat.java\jdt_ws\WEEK_1_8721193d\bin' 'NumberTwo'  
This is my first program  
I am on module ST4003CEM  
PS C:\Users\nawad\OneDrive\Desktop\WEEK_1>
```

3. Write a program that prints the following output:

A "quoted" String is

'much' better if you learn

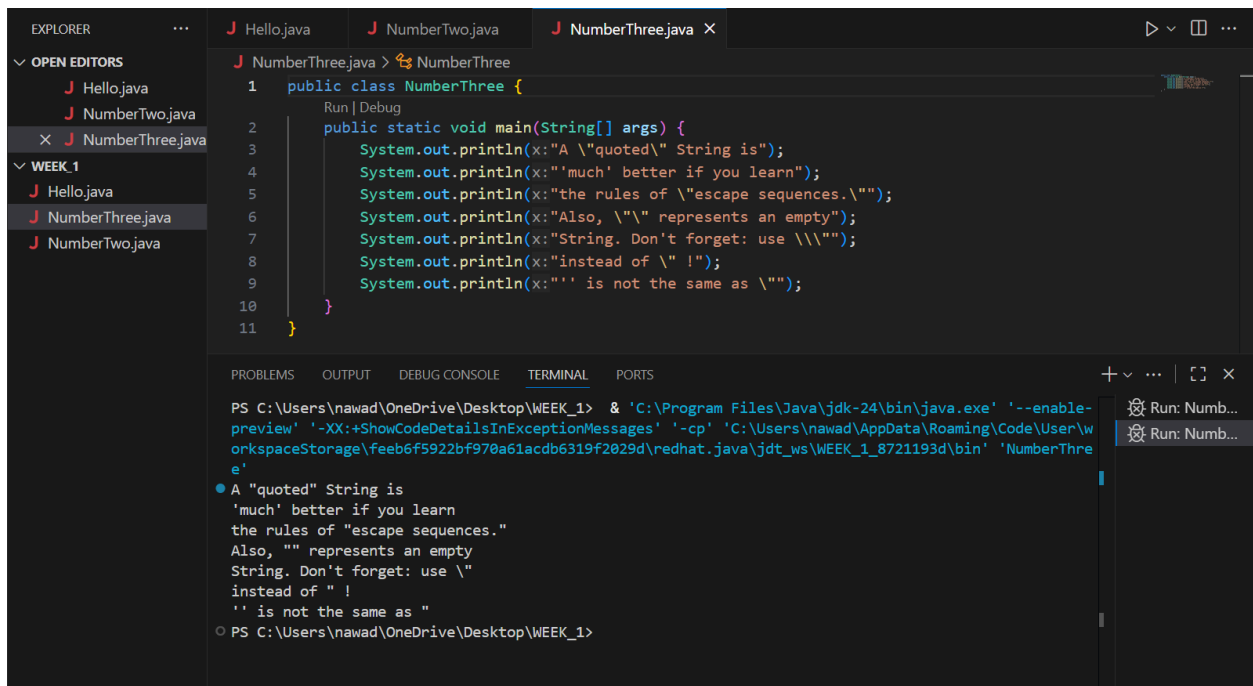
the rules of "escape sequences."

Also, "" represents an empty

String. Don't forget: use \"

instead of " !

" is not the same as "



The screenshot shows an IDE with three tabs: Hello.java, NumberTwo.java, and NumberThree.java. The NumberThree.java tab is active, displaying the following code:

```
1 public class NumberThree {  
2     public static void main(String[] args) {  
3         System.out.println("A \"quoted\" String is");  
4         System.out.println("'much' better if you learn");  
5         System.out.println("the rules of \"escape sequences.\"");  
6         System.out.println("Also, \"\" represents an empty");  
7         System.out.println("String. Don't forget: use \\\"\\\"");  
8         System.out.println("instead of \" !");  
9         System.out.println("' is not the same as \"");  
10    }  
11 }
```

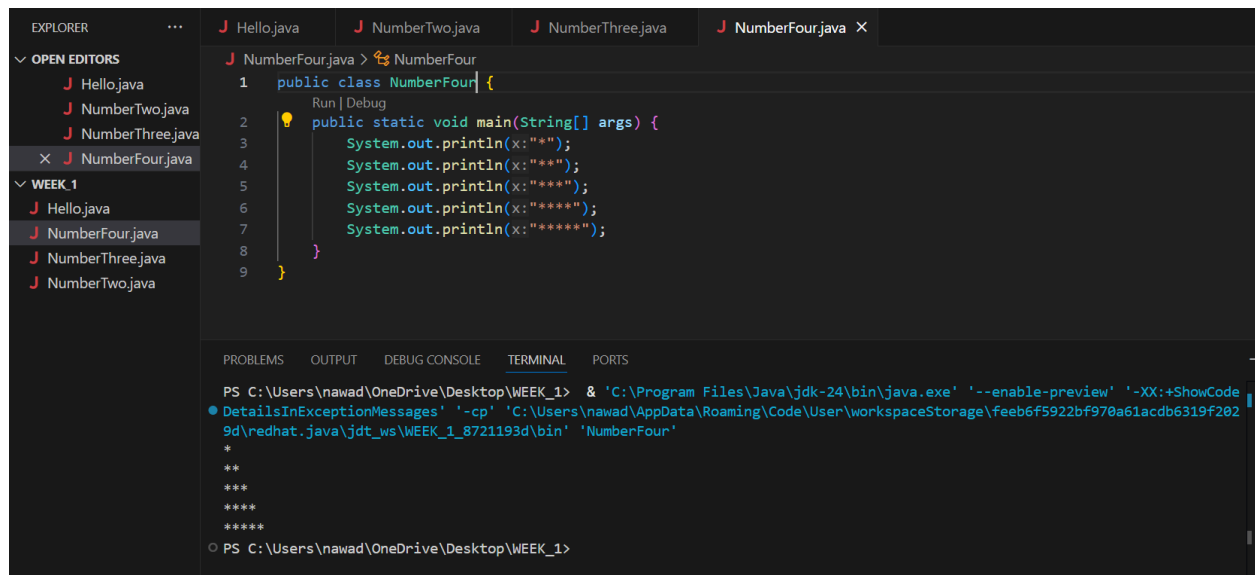
The terminal output at the bottom shows the command executed and the resulting output:

```
PS C:\Users\nawad\OneDrive\Desktop\WEEK_1> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\nawad\AppData\Roaming\Code\User\workspaceStorage\feeb6f5922b970a61acdb6319f2029d\redhat.java\jdt_ws\WEEK_1_8721193d\bin' 'NumberThree'  
● A "quoted" String is  
  'much' better if you learn  
  the rules of "escape sequences."  
  Also, "" represents an empty  
  String. Don't forget: use \"  
  instead of " !  
  ' is not the same as "  
○ PS C:\Users\nawad\OneDrive\Desktop\WEEK_1>
```

4. WAP that prints the following pattern:

*

**



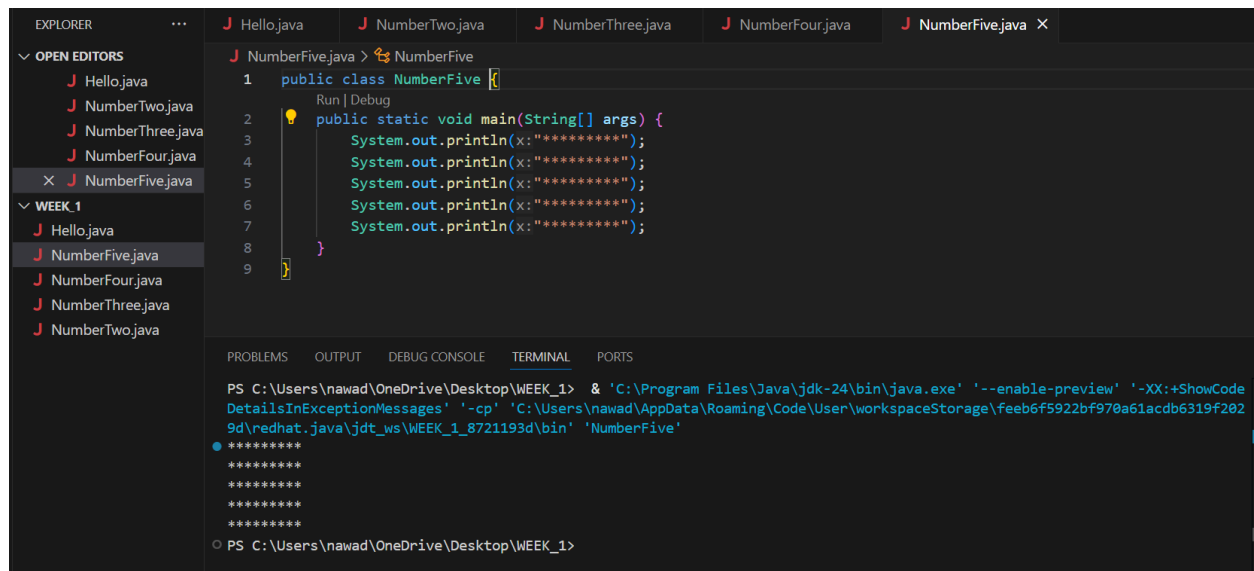
The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'WEEK_1' with files 'Hello.java', 'NumberFour.java', 'NumberThree.java', and 'NumberTwo.java'. The code editor shows the 'NumberFour.java' file with the following code:

```
1 public class NumberFour {  
2     public static void main(String[] args) {  
3         System.out.println(x:"*");  
4         System.out.println(x:"**");  
5         System.out.println(x:"***");  
6         System.out.println(x:"****");  
7         System.out.println(x:"*****");  
8     }  
9 }
```

The terminal output shows the execution of the program, printing the pattern of asterisks:

```
PS C:\Users\nawad\OneDrive\Desktop\WEEK_1> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCode  
DetailsInExceptionMessages' '-cp' 'C:\Users\nawad\AppData\Roaming\Code\User\workspaceStorage\feeb6f5922bf970a61acdb6319f202  
9d\redhat.java\jdt_ws\WEEK_1_8721193d\bin' 'NumberFour'  
*  
**  
***  
****  
*****  
PS C:\Users\nawad\OneDrive\Desktop\WEEK_1>
```

5. WAP prints the following pattern:



The screenshot shows an IDE with the Explorer on the left, a list of open editors in the middle, and a code editor on the right. The code editor displays a Java class named `NumberFive` with a `main` method that prints five lines of asterisks. The terminal at the bottom shows the command to run the program and the resulting output.

```
public class NumberFive {  
    public static void main(String[] args) {  
        System.out.println(x:"*****");  
        System.out.println(x:"*****");  
        System.out.println(x:"*****");  
        System.out.println(x:"*****");  
        System.out.println(x:"*****");  
    }  
}
```

```
PS C:\Users\nawad\OneDrive\Desktop\WEEK_1> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\nawad\AppData\Roaming\Code\User\workspaceStorage\feeb6f5922bf970a61acdb6319f2029d\redhat.java\jdt_ws\WEEK_1_8721193d\bin' 'NumberFive'  
*****  
*****  
*****  
*****  
*****  
PS C:\Users\nawad\OneDrive\Desktop\WEEK_1>
```

6. WAP that prints the following output:

A well-formed Java program has a main method with { and } braces.

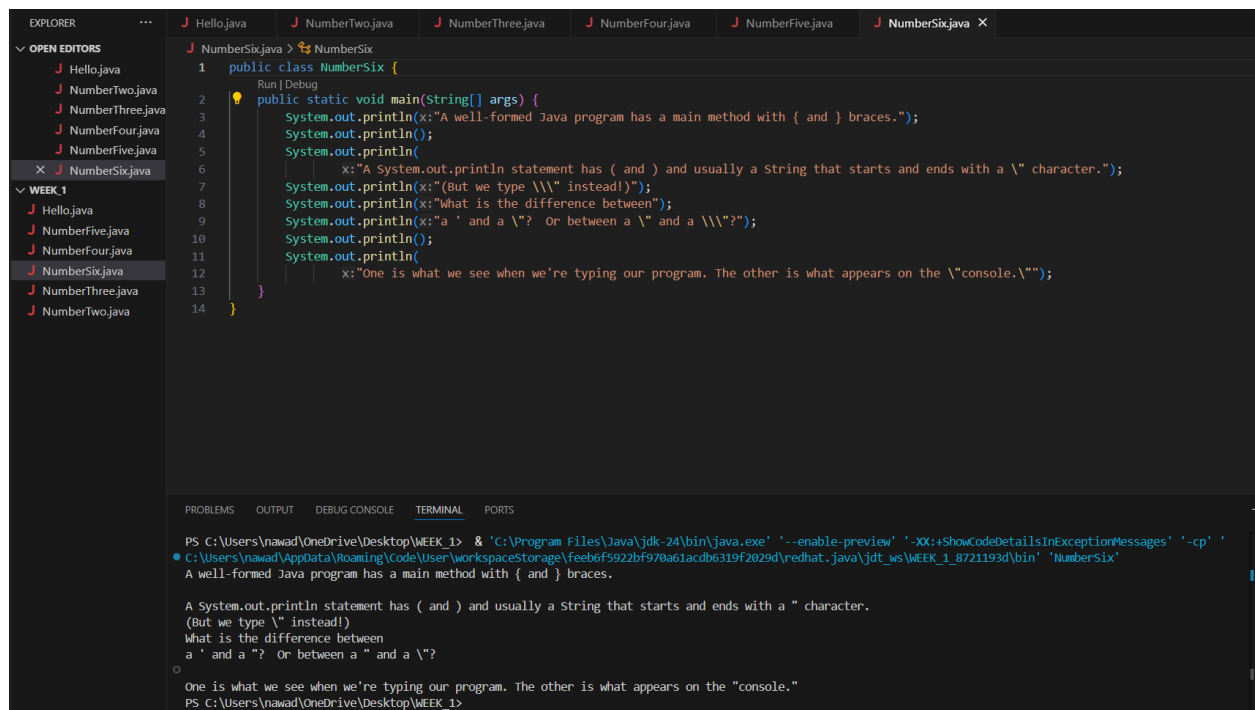
A `System.out.println` statement has (and) and usually a String that starts and ends with a " character.

(But we type \" instead!)

What is the difference between

a ' and a "? Or between a " and a \"?

One is what we see when we're typing our program. The other is what appears on the "console."



The screenshot shows an IDE with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'WEEK_1' with several Java files. The code editor displays the content of 'NumberSix.java', which is a public class with a main method. The main method contains several System.out.println statements that print out the answers to the questions posed in the text above. The terminal at the bottom shows the command to run the program and the resulting output, which matches the text in the main method.

```
1 public class NumberSix {
2     public static void main(String[] args) {
3         System.out.println(x:"A well-formed Java program has a main method with { and } braces.");
4         System.out.println();
5         System.out.println(
6             x:"A System.out.println statement has ( and ) and usually a String that starts and ends with a \" character.");
7         System.out.println(x: "(But we type \\\" instead!)");
8         System.out.println(x:"What is the difference between");
9         System.out.println(x:"a ' and a \"? Or between a \" and a \\\"?");
10        System.out.println();
11        System.out.println(
12            x:"One is what we see when we're typing our program. The other is what appears on the \"console.\"");
13    }
14 }
```

PS C:\Users\nawad\OneDrive\Desktop\WEEK_1> & "C:\Program Files\Java\jdk-24\bin\java.exe" "-enable-preview" "-XX:+ShowCodeDetailsInExceptionMessages" "-cp" "C:\Users\nawad\AppData\Roaming\code\User\workspaceStorage\feeb6f5922bf978a61acdb6319f2029d\redhat.java\jdt_ws\WEEK_1_8721193d\bin" "NumberSix"

A well-formed Java program has a main method with { and } braces.

A System.out.println statement has (and) and usually a String that starts and ends with a " character.

(But we type \" instead!)

What is the difference between

a ' and a "? Or between a " and a \"?

One is what we see when we're typing our program. The other is what appears on the "console."

PS C:\Users\nawad\OneDrive\Desktop\WEEK_1>