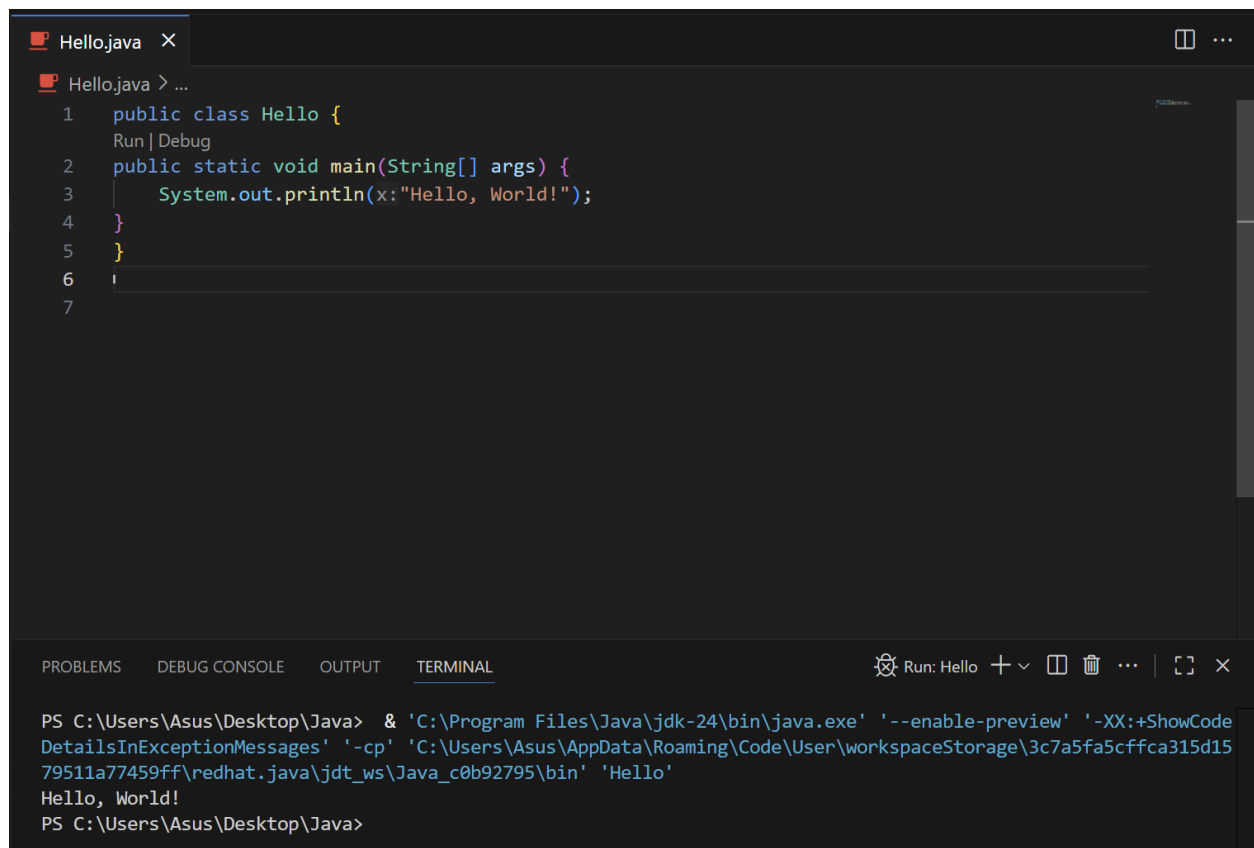


Jharana Adhikari

Computing (38)

Task to Do

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



The screenshot shows an IDE window titled 'Hello.java'. The code editor contains the following Java code:

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

Below the code editor is a terminal window with the following output:

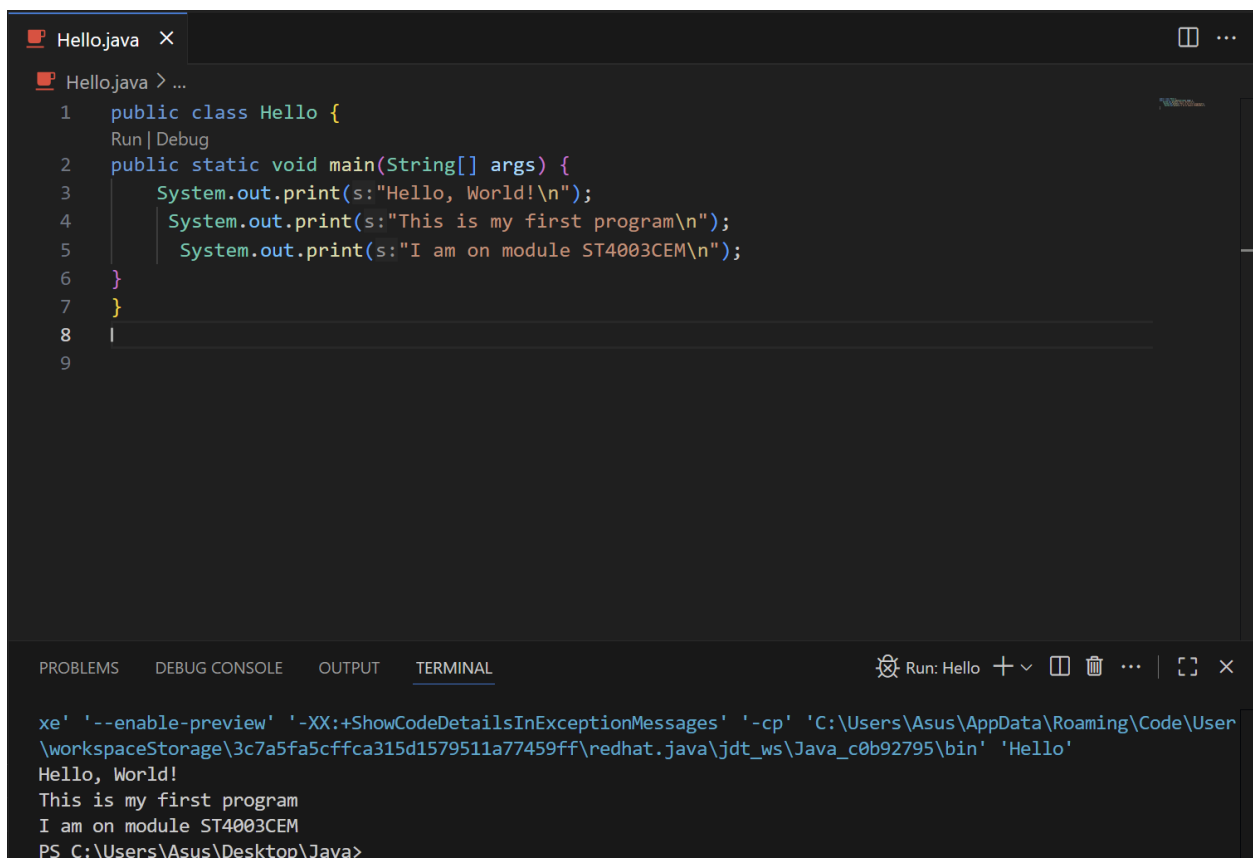
```
PS C:\Users\Asus\Desktop\Java> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Asus\AppData\Roaming\Code\User\workspaceStorage\3c7a5fa5cffa315d1579511a77459ff\redhat.java\jdt_ws\Java_c0b92795\bin' 'Hello'  
Hello, World!  
PS C:\Users\Asus\Desktop\Java>
```

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 window with a file named 'Hello.java'. The code in the editor is as follows:

```
1 public class Hello {  
2     Run | Debug  
3     public static void main(String[] args) {  
4         System.out.print(s:"Hello, World!\n");  
5         System.out.print(s:"This is my first program\n");  
6         System.out.print(s:"I am on module ST4003CEM\n");  
7     }  
8 }  
9
```

Below the editor is a terminal window with the following output:

```
xe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Asus\AppData\Roaming\Code\User\workspaceStorage\3c7a5fa5cffca315d1579511a77459ff\redhat.java\jdt_ws\Java_c0b92795\bin' 'Hello'  
Hello, World!  
This is my first program  
I am on module ST4003CEM  
PS C:\Users\Asus\Desktop\Java>
```

3. Write a program that prints the following output:

A "quoted" String is

'much' better if you learn

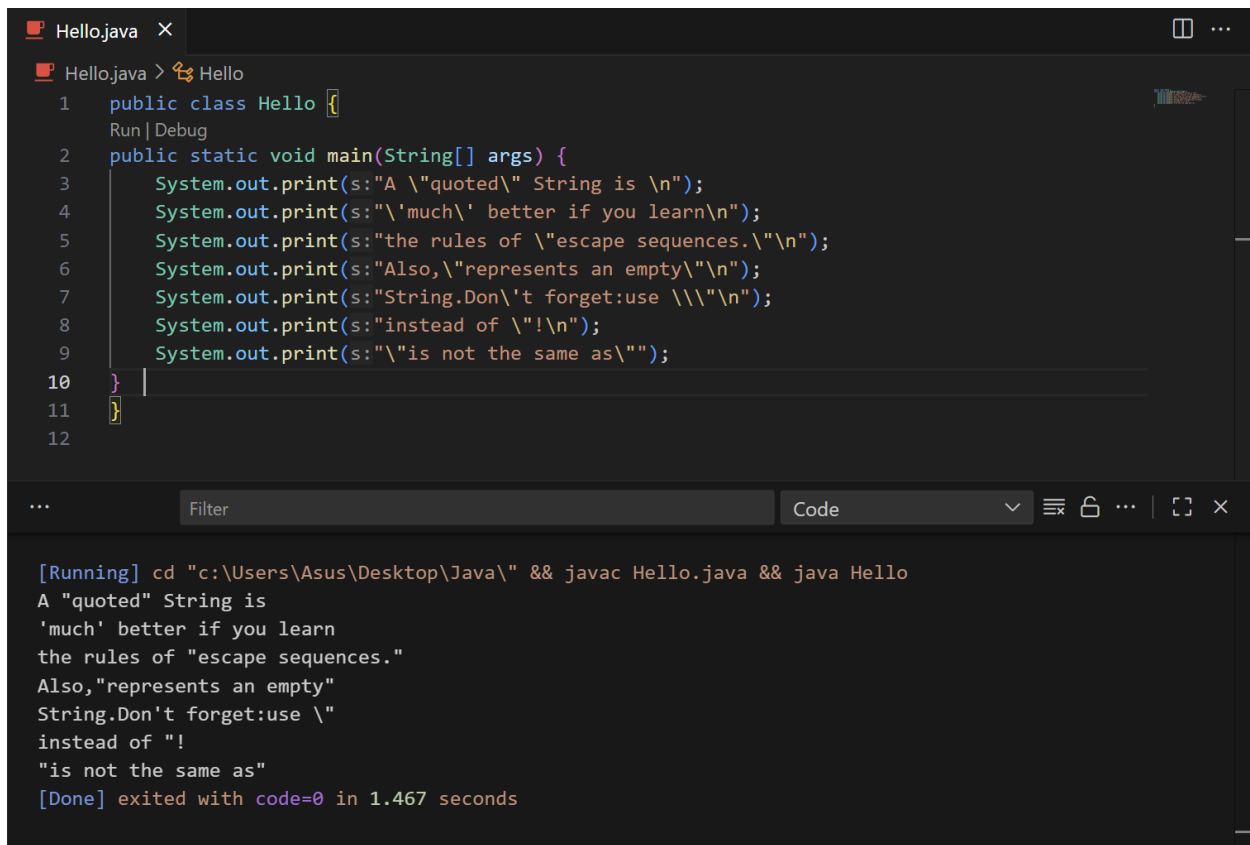
the rules of "escape sequences."

Also, "" represents an empty

String. Don't forget: use \"

instead of " !

" is not the same as "



The screenshot shows an IDE window titled 'Hello.java'. The code defines a class 'Hello' with a 'main' method that prints several lines of text using 'System.out.print'. The output is displayed in the console below the code editor. The output matches the text provided in the previous blocks, demonstrating the use of escape sequences for quotes and backslashes.

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         System.out.print(s:"A \"quoted\" String is \n");  
4         System.out.print(s:"'much' better if you learn\n");  
5         System.out.print(s:"the rules of \"escape sequences.\\\"\\n");  
6         System.out.print(s:"Also,\"represents an empty\"");  
7         System.out.print(s:"String.Don't forget:use \\\"\\n");  
8         System.out.print(s:"instead of \"!\n");  
9         System.out.print(s:"\" is not the same as\"");  
10    }  
11 }  
12
```

[Running] cd "c:\Users\Asus\Desktop\Java\" && javac Hello.java && java Hello  
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 "  
[Done] exited with code=0 in 1.467 seconds

#### 4. WAP that prints the following pattern:

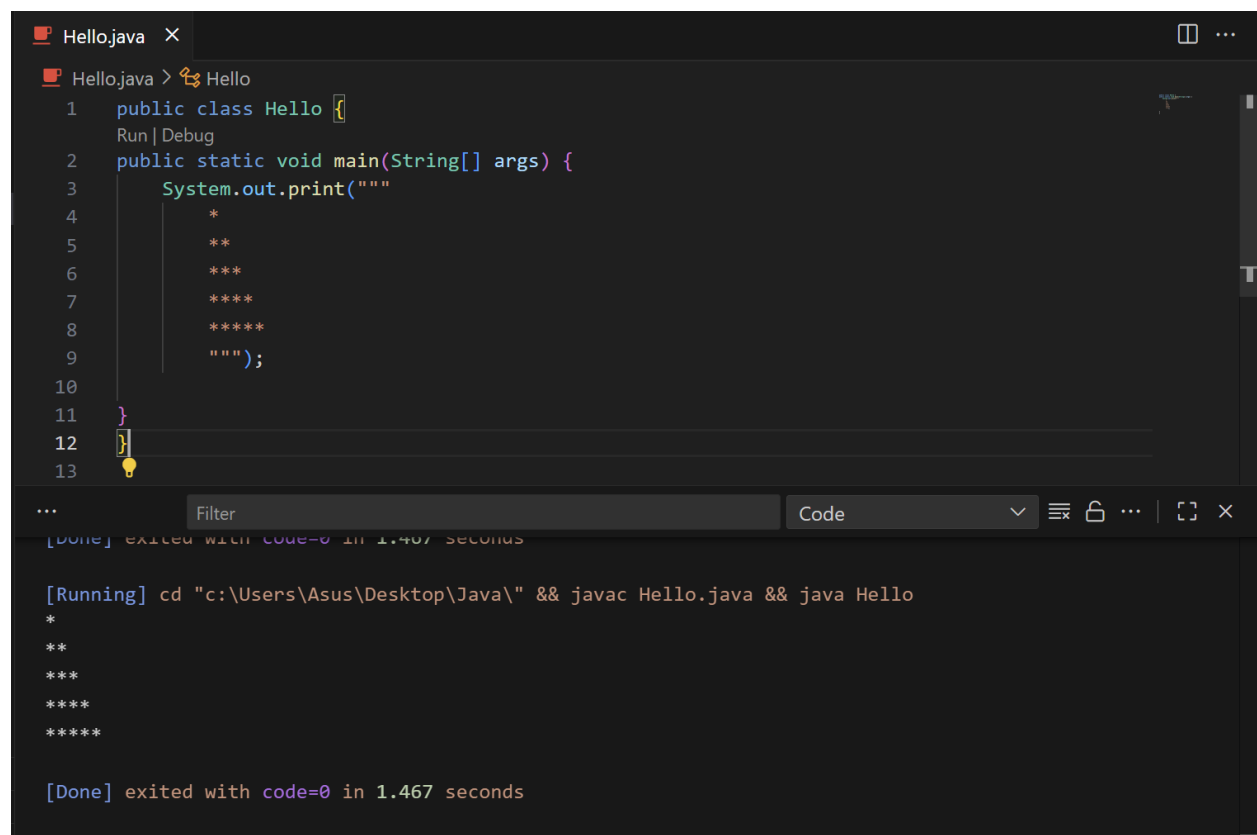
\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*



The screenshot shows an IDE window titled 'Hello.java'. The code defines a class 'Hello' with a 'main' method that prints a pattern of stars. The output window shows the command 'cd "c:\Users\Asus\Desktop\Java\" && javac Hello.java && java Hello' and the resulting star pattern.

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         System.out.print("  
4             *  
5             **  
6             ***  
7             ****  
8             *****  
9             """);  
10    }  
11 }  
12  
13
```

[Done] exited with code=0 in 1.467 seconds

[Running] cd "c:\Users\Asus\Desktop\Java\" && javac Hello.java && java Hello

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

[Done] exited with code=0 in 1.467 seconds

## 5. WAP prints the following pattern:

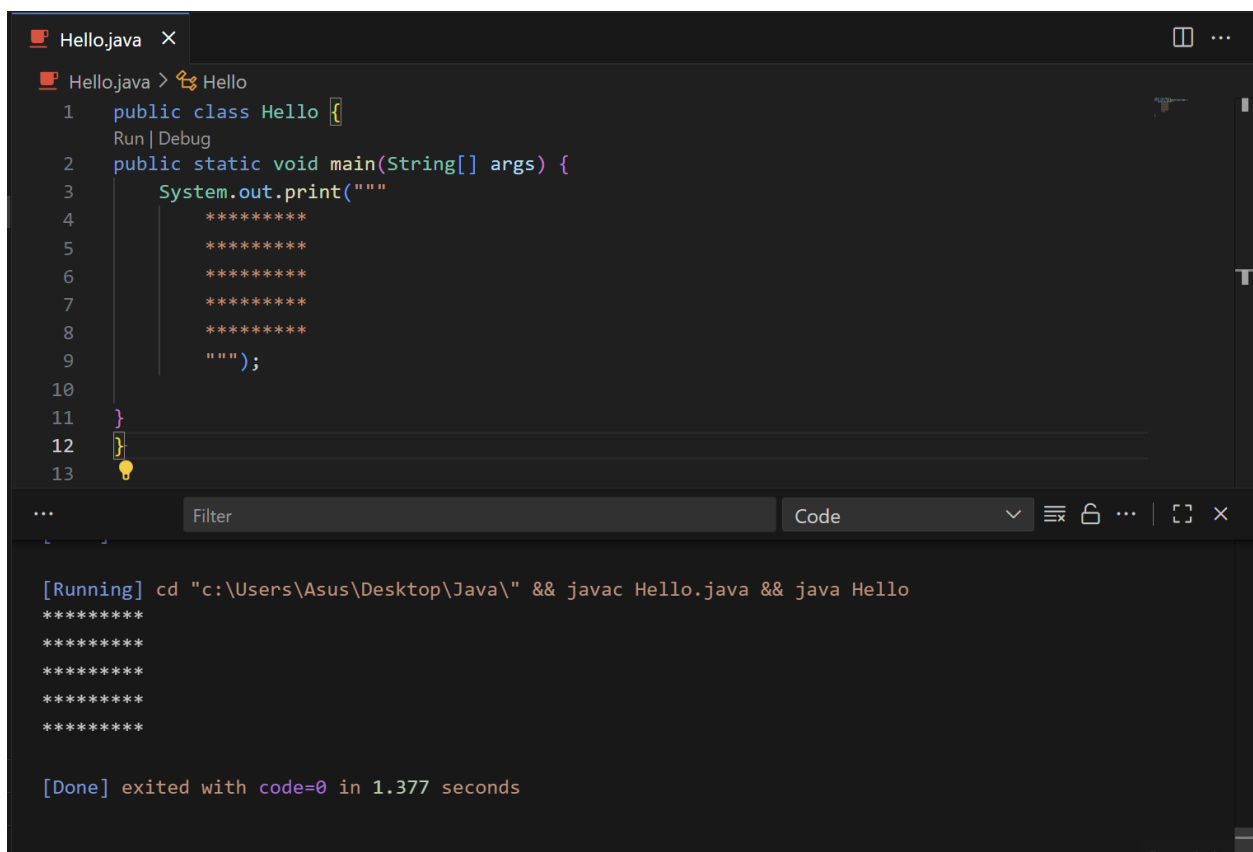
\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*



The screenshot shows an IDE window titled 'Hello.java'. The code defines a public class 'Hello' with a main method that prints five lines of asterisks. The output window at the bottom shows the command 'cd "c:\Users\Asus\Desktop\Java\" && javac Hello.java && java Hello' being executed, followed by the same five lines of asterisks, and a completion message: '[Done] exited with code=0 in 1.377 seconds'.

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         System.out.print(""  
4             *****  
5             *****  
6             *****  
7             *****  
8             *****  
9             """);  
10    }  
11 }  
12  
13
```

[Running] cd "c:\Users\Asus\Desktop\Java\" && javac Hello.java && java Hello  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
[Done] exited with code=0 in 1.377 seconds

6. WAP that prints the following output:

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

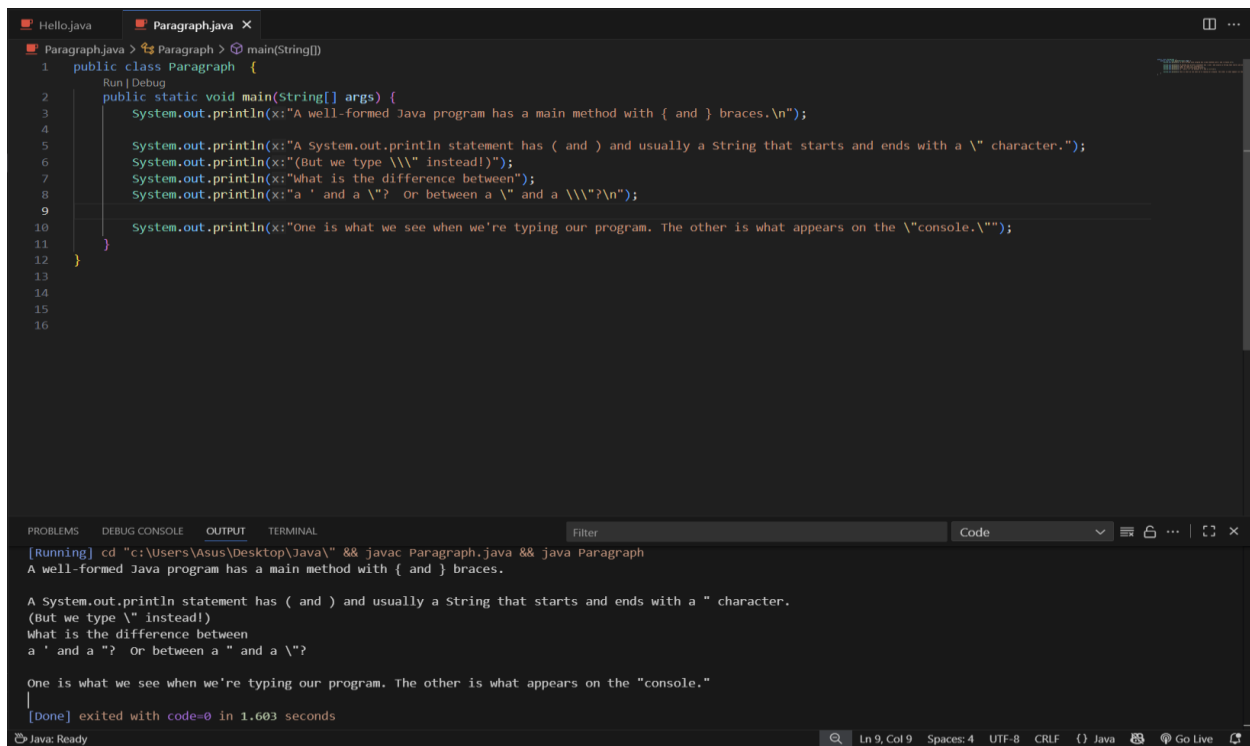
A System.out.println statement has ( and ) and usually a String that starts and ends with a " character.

(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 Java file named `Paragraph.java`. The code defines a `Paragraph` class with a `main` method. The `main` method contains five `System.out.println` statements that print the required output. The IDE's output window at the bottom shows the execution results, which match the expected output. The status bar at the bottom indicates the file is at line 9, column 9, with 4 spaces, UTF-8 encoding, and CRLF line endings.

```
1 public class Paragraph {
2     public static void main(String[] args) {
3         System.out.println("A well-formed Java program has a main method with { and } braces.\n");
4
5         System.out.println("A System.out.println statement has ( and ) and usually a String that starts and ends with a \" character.");
6         System.out.println("(But we type \" instead!)");
7         System.out.println("What is the difference between");
8         System.out.println("a ' and a "? Or between a \" and a \"?\n");
9
10        System.out.println("One is what we see when we're typing our program. The other is what appears on the \"console.\"");
11    }
12 }
13
14
15
16
```

[Running] cd "c:\Users\Asus\Desktop\Java\" && javac Paragraph.java && java Paragraph  
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."  
[Done] exited with code=0 in 1.603 seconds

Java: Ready | Ln 9, Col 9 | Spaces: 4 | UTF-8 | CRLF | () Java | Go Live

