



## **TASK 1**

**Prachi Silwal**

**BSc. Hons Computer Science with AI, Softwarica college of IT  
and E-commerce**

**ST40003CMD-Programming: Professional Practice**

**Prabisha Khadka**

**April 30, 2025**

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

The screenshot shows an IDE window titled "Java week1". The Explorer sidebar on the left shows a project named "JAVA WEEK1" containing a file "Hello.java". The main editor displays the following Java code:

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

Below the code editor, the "TERMINAL" tab is active, showing the execution of the program:

```
prachisilwal@Prachis-MacBook-Pro Java week1 % cd /Users/prachisilwal/Java\ week1 ; /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-24.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/prachisilwal/Library/Application\ Support/Code/User/workspaceStorage/c92c4f100a92de55de3272137b4a3a76/redhat.java/jdt_ws/Java\ week1_909f5707/bin Hello  
Hello World!  
prachisilwal@Prachis-MacBook-Pro Java week1 %
```

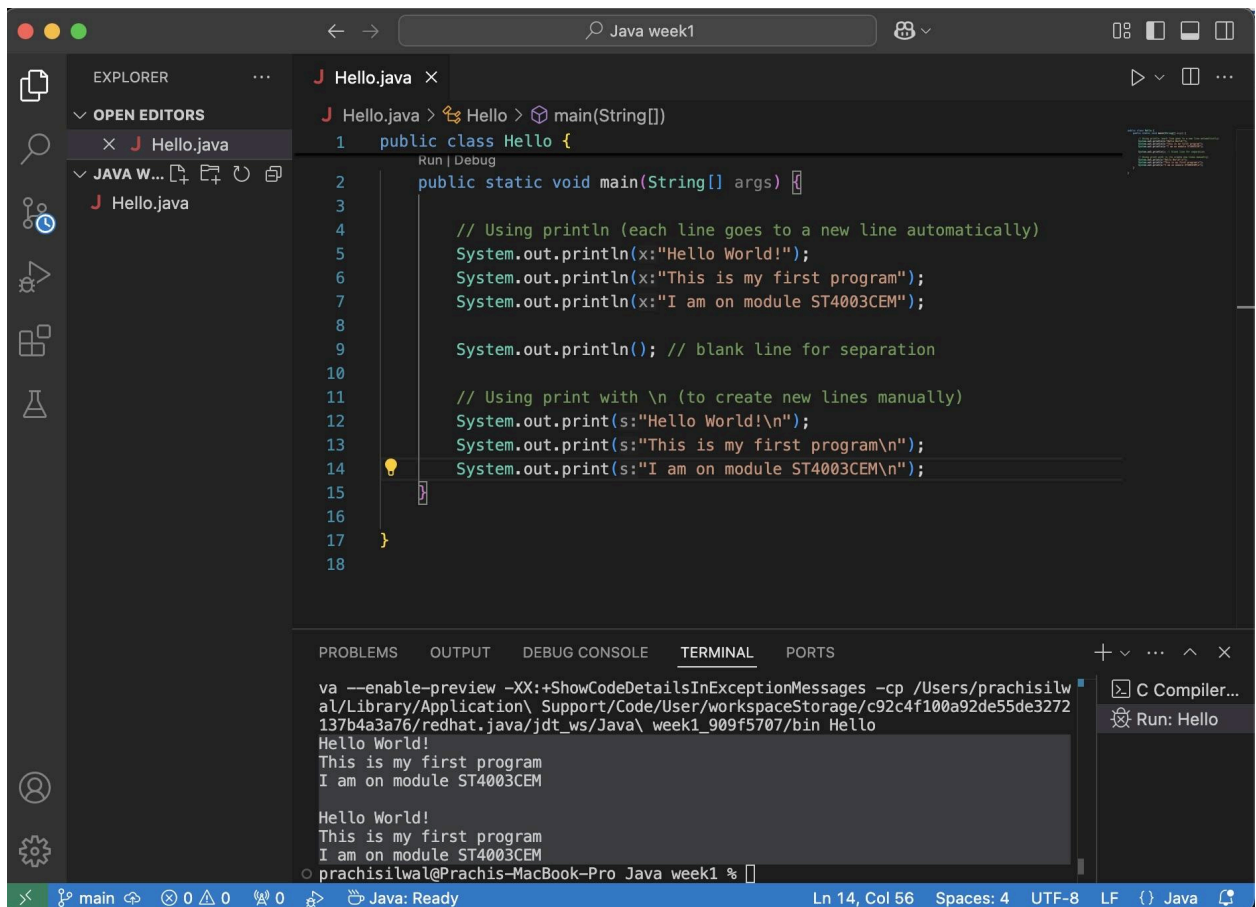
The status bar at the bottom indicates the current file is "main", there are no errors or warnings, and the Java runtime is ready. The cursor is at line 7, column 1.

## 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.



```
1 public class Hello {
2     public static void main(String[] args) {
3
4         // Using println (each line goes to a new line automatically)
5         System.out.println(x:"Hello World!");
6         System.out.println(x:"This is my first program");
7         System.out.println(x:"I am on module ST4003CEM");
8
9         System.out.println(); // blank line for separation
10
11        // Using print with \n (to create new lines manually)
12        System.out.print(s:"Hello World!\n");
13        System.out.print(s:"This is my first program\n");
14        System.out.print(s:"I am on module ST4003CEM\n");
15    }
16 }
17
18
```

Terminal Output:

```
va --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/prachisilw
al/Library/Application\ Support/Code/User/workspaceStorage/c92c4f100a92de55de3272
137b4a3a76/redhat.java/jdt_ws/Java\ week1_909f5707/bin Hello
Hello World!
This is my first program
I am on module ST4003CEM

Hello World!
This is my first program
I am on module ST4003CEM
prachisilwal@Prachis-MacBook-Pro Java week1 %
```

### 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 "

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

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
location\ Support\Code\User\workspaceStorage\c92c4f100a92de55de3272137b4a3a76/
redhat.java/jdt_ws/Java\ week1_9
09f5707/bin EscapeSequencesDemo
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 "
o prachisilwal@Prachis-MacBook-Pro Java week1 %
```

4. WAP that prints the following pattern:

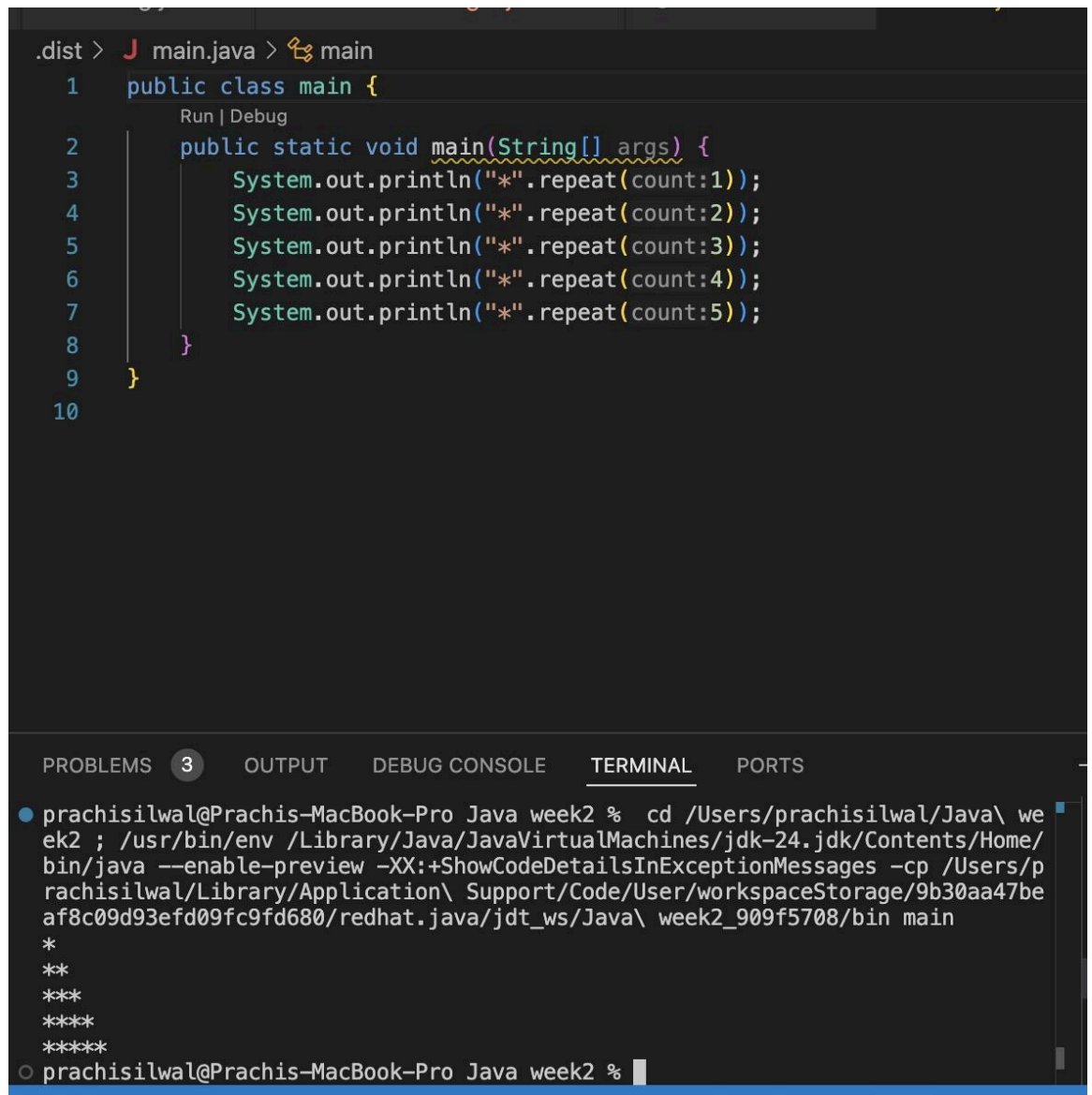
\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*



The screenshot shows an IDE with a Java file named `main.java`. The code defines a `main` method that prints five lines of asterisks, each line increasing by one asterisk. Below the code editor, the `TERMINAL` tab is active, showing the command to run the program and the resulting output, which matches the pattern shown in the previous blocks.

```
.dist > J main.java > main
1 public class main {
    Run | Debug
2     public static void main(String[] args) {
3         System.out.println("*".repeat(count:1));
4         System.out.println("*".repeat(count:2));
5         System.out.println("*".repeat(count:3));
6         System.out.println("*".repeat(count:4));
7         System.out.println("*".repeat(count:5));
8     }
9 }
10
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
prachisilwal@Prachis-MacBook-Pro Java week2 % cd /Users/prachisilwal/Java\ week2 ; /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-24.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/prachisilwal/Library/Application\ Support/Code/User/workspaceStorage/9b30aa47beaf8c09d93efd09fc9fd680/redhat.java/jdt_ws/Java\ week2_909f5708/bin main
*
**
***
****
*****
prachisilwal@Prachis-MacBook-Pro Java week2 %
```

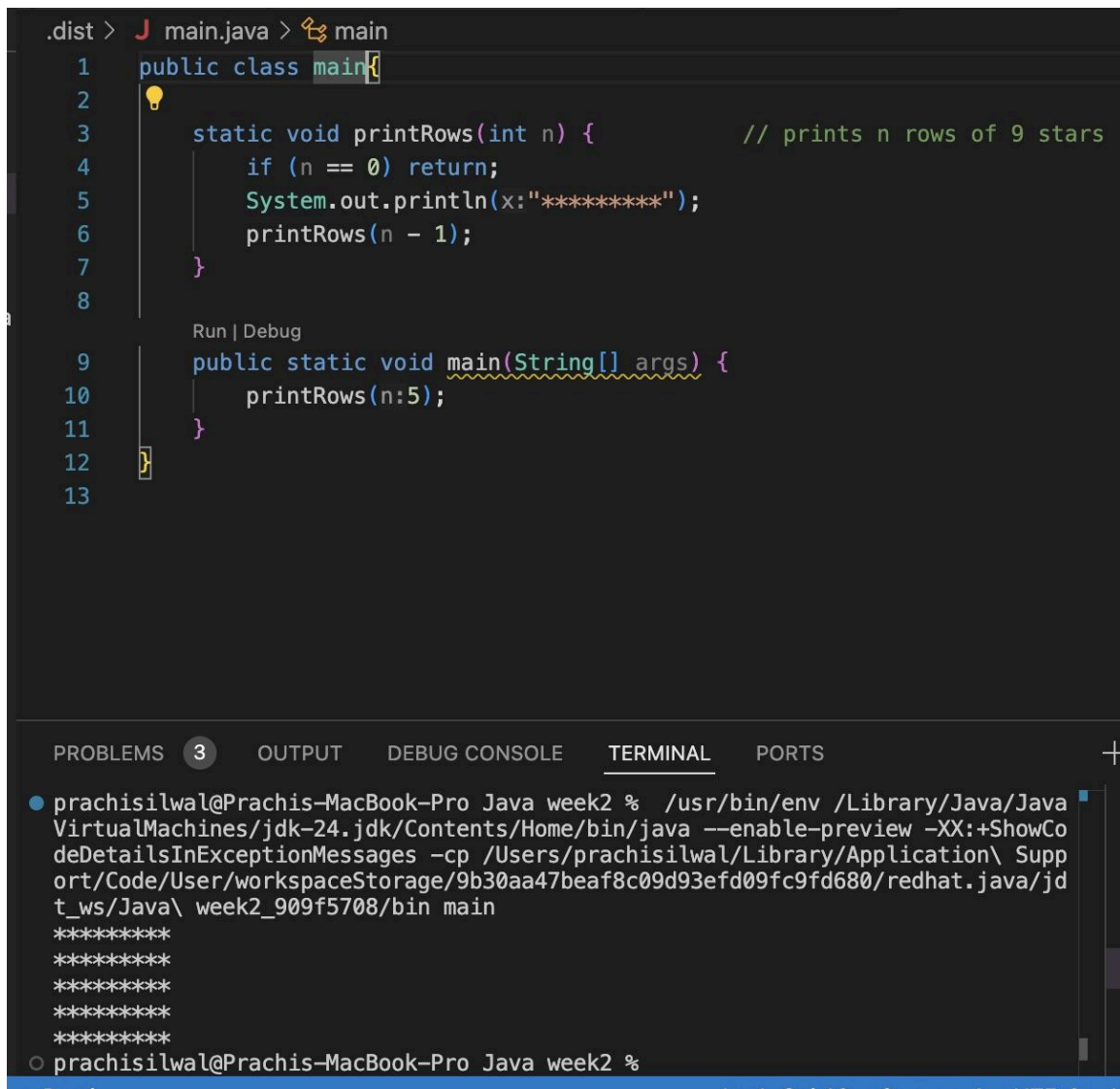
5. WAP prints the following pattern:

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*



The screenshot shows an IDE with a Java file named `main.java`. The code defines a `main` class with a `printRows` method that prints a specified number of rows of stars. The `main` method calls `printRows(5)`. Below the code editor, the `TERMINAL` tab shows the command to run the program, which successfully prints the pattern of five rows of stars.

```
.dist > J main.java > main
1 public class main{
2
3     static void printRows(int n) {           // prints n rows of 9 stars
4         if (n == 0) return;
5         System.out.println(x:"*****");
6         printRows(n - 1);
7     }
8
9     public static void main(String[] args) {
10        printRows(n:5);
11    }
12 }
13
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
prachisilwal@Prachis-MacBook-Pro Java week2 % /usr/bin/env /Library/Java/Java
VirtualMachines/jdk-24.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCo
deDetailsInExceptionMessages -cp /Users/prachisilwal/Library/Application\ Supp
ort/Code/User/workspaceStorage/9b30aa47beaf8c09d93efd09fc9fd680/redhat.java/jd
t_ws/Java\ week2_909f5708/bin main
*****
*****
*****
*****
*****
prachisilwal@Prachis-MacBook-Pro Java week2 %
```

## 6. WAP that prints the following output:

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

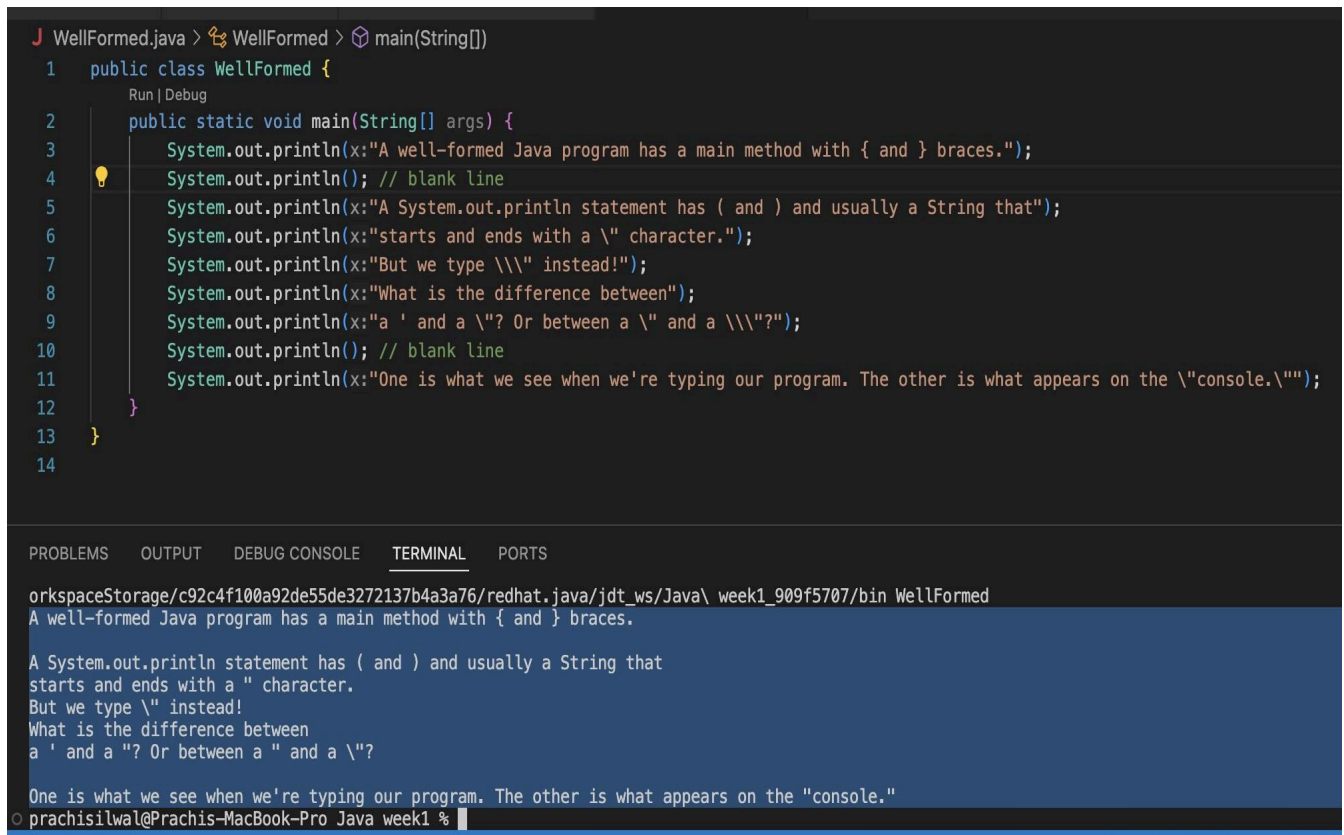
A System.out.println statement has ( and ) and usually a String that starts and ends with a " character.

(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 WellFormed.java. The code defines a public class WellFormed with a main method. The main method contains several System.out.println statements that print the required output. The IDE's output window shows the execution results, which match the expected output. The terminal window shows the command to run the program.

```
WellFormed.java > WellFormed > main(String[])
1 public class WellFormed {
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(); // blank line
5         System.out.println(x:"A System.out.println statement has ( and ) and usually a String that");
6         System.out.println(x:"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(); // blank line
11        System.out.println(x:"One is what we see when we're typing our program. The other is what appears on the \"console.\"");
12    }
13 }
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

orkspaceStorage/c92c4f100a92de55de3272137b4a3a76/redhat.java/jdt\_ws/Java\ week1\_909f5707/bin WellFormed

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."

prachisilwal@Prachis-MacBook-Pro Java week1 %