CMM024 Object Oriented Programming

Practical Session: 1

1. Write a code to display a single line of text with multiple statements

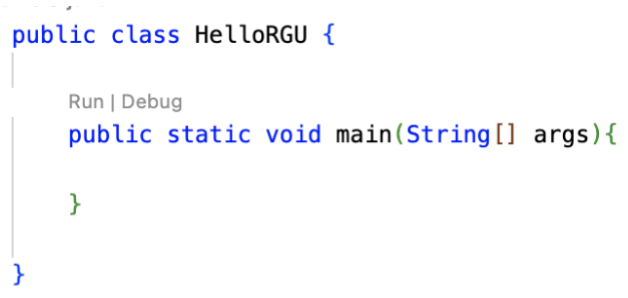
The string "Welcome to Object Oriented Programming!" can be displayed in several ways. After creating a Class called **HelloRGU**, use two print statements to produce the output shown below. The first statement uses **System.out** 's method **print** to display the first string. The second one use the **println** method instead of **print** to display the second one.

Note:

Each print or **println** statement resumes displaying characters from where the last print or **println** statement stopped displaying characters. Unlike **println**, after displaying its argument which adds an end of line, **print** does not position the output cursor at the beginning of the next line in the command window—the next character the program displays will appear immediately after the last character that print displays.

Tasks:

1) Create a **HelloRGU** java file



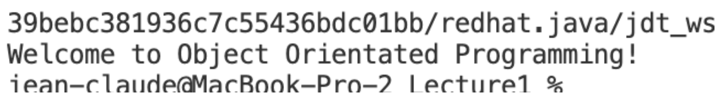
```java
public class HelloRGU {

    public static void main(String[] args){

    }

}
```

2) Create a **main** function
3) Create a string, name it **st1** and assign **"Welcome to "**
4) Create another string, name it **st2** and assign **"Object Orientated Programming"**
5) Use **print** display, the first string

6) Use **println** display the second string

7) Run the program

Output:



```
39bebc381936c7c55436bdc01bb/redhat.java/jdt_ws
Welcome to Object Orientated Programming!
jean-claude@MacBook-Pro-2 Lecture1 %
```

2. Write a code for Displaying Multiple Lines of Text with a Single Statement

Note:

A single statement can display multiple lines by using newline characters, which indicate to **System.out** 's **print** and **println** methods when to position the output cursor at the beginning of the next line in the command window. Like blank lines, space characters and tab characters, newline characters are whitespace characters. In this example, the code should output four lines of text, using newline characters to determine when to begin each new line.

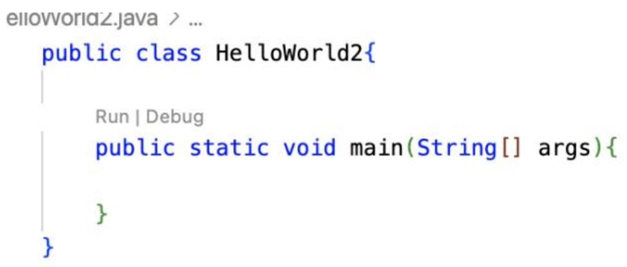
Explanations:

Normally, the characters in a string are displayed *exactly* as they appear in the double quotes. However, the paired characters \ and n (repeated four times in the statement) do *not* appear on the screen.

The **backslash** (\) is an **escape character**, which has special meaning to **System.out** 's print and **println** methods. When a backslash appears in a string, Java combines it with the next character to form an **escape sequence**, i.e., \n which tells Java it is a newline character. When a newline character appears in a string being output with **System.out**, the newline character causes the screen's output cursor to move to the beginning of the next line in the command window.

Tasks:

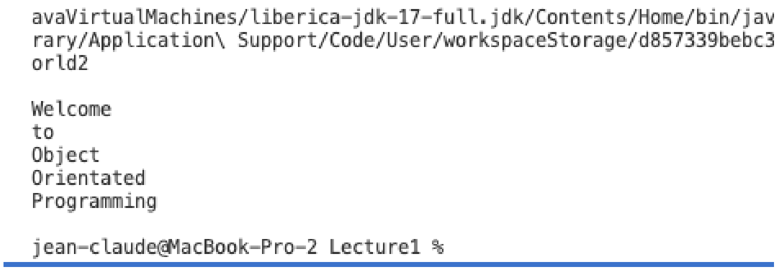
1) Create a file called HelloWorld2 and create the main function



```
elloWorld2.java > ...
public class HelloWorld2{

    Run | Debug
    public static void main(String[] args){

    }
}
```

2) Create code using the **System.out.println** method to display in the terminal console the output shown below by adding the "\n" newline characters in the right position.

Output:



```
avaVirtualMachines/liberica-jdk-17-full.jdk/Contents/Home/bin/jav
rary/Application\ Support/Code/User/workspaceStorage/d857339bebc3
orld2

Welcome
to
Object
Orientated
Programming

jean-claude@MacBook-Pro-2 Lecture1 %
```

3. Write Declarations, statements that accomplish each of the following tasks:

The purpose of this exercise is to create a program will calculate the product of three integers.

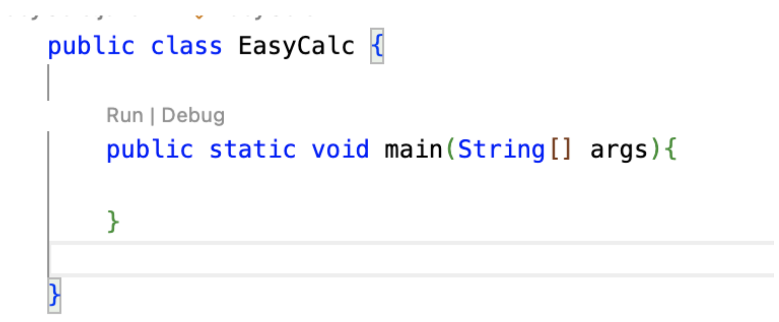
Note:

1. When displaying some value using the **System.out.printf** you must use %.3f, which must be inserted in the right place. Here 3 means 3 decimal values. Below is an example.

```
System.out.printf("Some text about variable: %.3f\n", variable);
```

Tasks:

2. Create a new java file called EasyCalc, and create a main method



```
public class EasyCalc {

    Run | Debug
    public static void main(String[] args){

    }

}
```

3. Create a Scanner called input that reads values from the standard input.

4. Declare the variables x, y, z and result to be of type double.

5. Display a message in the console to inform user to enter the first number.

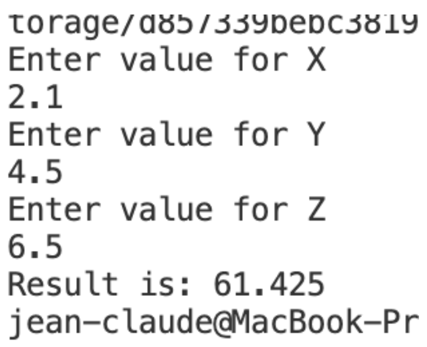
6. Using the scanner input, prompt the user to enter the first integer and assign that value in the variable x.

7. Repeat the same process for y and z.

8. Compute the product of the three double values contained in variables **x**, **y** and **z**, and assign the result to the variable **result**.

9. Display the value of result (3 decimal point) using the **System.out.printf** method to display the message **"Result is: "** followed by the value of the variable result.

<u>Output:</u>



```
torage/d85/339bebc3819
Enter value for X
2.1
Enter value for Y
4.5
Enter value for Z
6.5
Result is: 61.425
jean-claude@MacBook-Pr
```