

Aε01 : Fun with Hello World!

Created on Monday, July 19, 2013

Updated on July 19, 2013

Authored by:

Nicholas ‘Timethor’ Rich

Contents

Overview	2
Background	2
Part 1: Basic Hello World!	3
Program	3
Requirements	3
Notice!	3
Example Output	3
Part 2: Hello World! on separate lines	3
Program	3
Requirements	3
Example Output	3
Part 3: Hello World! using formatting	4
Program	4
Requirements	4
Example Output	4
Part 4: Hello World! in a window	4
Program	4
Requirements	4
Example Output	4
BONUS: More <code>printf()</code>	5
Program	5
Guidelines	5
Hints	5
Example Output	5

Overview

This assignment will be an exercise in getting used to Java output. You will be writing one of the most popular introductory programs in the world. Hello World!

Background

There are many methods in Java that will help your program talk to a user. There are three main versions that output to the console and one that creates a small window with an Ok button so you can close it.

Console Output

```
System.out.print("Hello there");
```

This one outputs what is in between the parenthesis without going to a new line after.

```
System.out.println("Hello there");
```

This one outputs what is in between the parenthesis and then goes to the next line.

```
System.out.printf("%s", "Hello there");
```

This one allows you to format your output. For each %s, `printf()` expects a new parameter. For example: `System.out.printf("%s %s", "Hello", "there");` prints the same as the one above. Notice there is a space between the two %s's this time. Essentially, the %s's are replaced with each of the following inputs. So, the first %s is replaced with "Hello" and the second is replaced with "there". Interesting right?

Windowed Output

```
JOptionPane.showMessageDialog(null, "Hello there")
```

This one creates a small window that displays the message given. Running this multiple times will open multiple windows.

Part 1: Basic Hello World!

Program

For this first part, you will be using `System.out.print()`; for your output.

Requirements

You need to use a different `print()` for each word and punctuation mark. In total that is three `print()`s.

Notice!

See how using `print()` puts everything on a single line? Next we will explore how to put text on a new line.

Example Output

```
Hello World!
```

Part 2: Hello World! on separate lines

Program

Now, add to what you already have. Use `System.out.println()`; for new your output.

Requirements

You need to use a different `println()` for each word and punctuation mark. In total that is three `println()`s to add.

Also, You should have a line from the previous part that looks like:

```
System.out.print("!");
```

change it to:

```
System.out.print("\n");
```

The “\n” that you added will turn `print()` into `println()`

Example Output

```
Hello World!  
Hello  
World  
!
```

Part 3: Hello World! using formatting

Program

Again, add to what you already have. This time, use `System.out.printf()`; for new your output.

Requirements

Just use one `printf()` for this part. This time for the format part, use three `%s`’s and give `printf()` the extra input `"Hello", "World", "!"`. See the background section if you are confused.

Example Output

```
Hello World!  
Hello  
World  
!  
5 Hello World!
```

Part 4: Hello World! in a window

Program

Again, add to what you already have. This time, use `JOptionPane.showMessageDialog()` for new your output.

if you are not using netbeans or another IDE, you will need to add an import at the top of the file like this:
`import javax.swing.JOptionPane;`

Requirements

This one can be simple. Just use `JOptionPane` to make a window that says Hello World!.

The `"\n"` that you added will turn `print()` into `println()`

Example Output

Console:

```
Hello World!  
Hello  
World  
!  
5 Hello World!
```

Window:

```
Hello World!
```

BONUS: More printf()

Program

Enough of Hello World!, let’s have some real fun! Comment out everything you have so far. This part will take a bit of thinking if you have never done anything like this before, but nothing you can’t accomplish!

Guidelines

This part is more open-ended for you to play around as you please, so rather than requirements here we just have guidelines. Write a short story of about 6 lines or copy paste from something of your choice. Find a way to format it so that it looks good when the console prints it i.e. It is wrapped to a certain number of characters. Try using `printf` to do this. So you get used to using it as it will be important for later assignments.

Hints

- The “\n” that you added to `print()` can also be added to `printf()` in the format string like “%s\n” to get the same effect.
- You will need to make a String variable to store your story.
- Strings have the methods `length()` and `substring()` that can be helpful for wrapping the story.
- Strings also have the methods `startsWith()` and `endsWith()` that can be helpful for adding a dash (“-”) if needed at the end of a line.
- You want to create a loop that will end when your story has been broken into wrapped lines and printed.

Example Output

Console:

```
5 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut  
   purus elit, vestibulum ut, placerat ac, adipiscing vitae, f-  
   elis. Curabitur dictum gravida mauris. Nam arcu libero, nonu-  
   mmy eget, consectetur id, vulputate a, magna. Donec vehicul-  
   a augue eu neque. Pellentesque habitant morbi tristique sene-  
   ctus et netus et malesuada fames ac turpis egestas.
```