STUDY GUIDE 1: Hello World!

AP Computer Science - WHS Mulvaney Name _____ Per _____

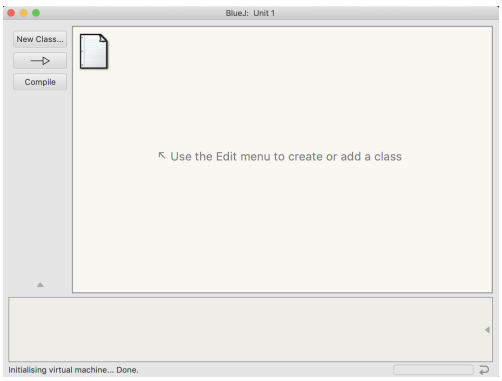
Resources

Lecture Slides	Video Lectures	Textbook Section
- 1A: Hello World!	- Hello World! - Print Statements and Escape Sequences	- 1.2 And Now-Java
PracticeIt (Building Java Programs 4th Edition, Chapter 1) - Self Check: 1.8, 1.13, 1.15 - Exercise: 1.1, 1.4		

Task 1: Hello World

When learning a new programming language, it is tradition to use that language to print the words 'Hello World' on the screen. Some languages require a lot of characters to perform this task (Java) and some require less (Python).

1. Complete the BlueJ and Eclipse setup procedures. You will know this is done if BlueJ shows a blank workspace window.



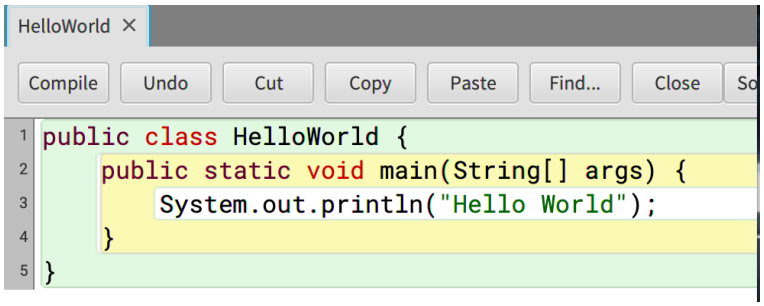
2. Click on the New Class Button.

3. Name your class HelloWorld
    a. Note the lack of space between the two words and the capitalization of both the 'H' and the 'W'.

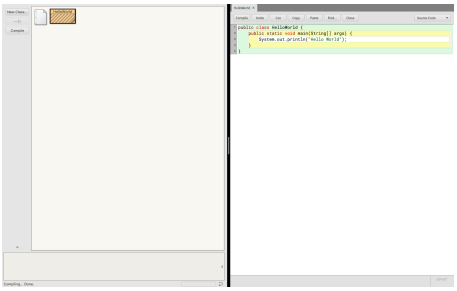
4. A tan 'HelloWorld' box with diagonal slashes now shows in our workspace. Double click this box to open the editor window.

5. The makers of BlueJ, the University of Kent, were kind enough to provide us a lot of code for all our new classes. How nice of them. Delete it all.

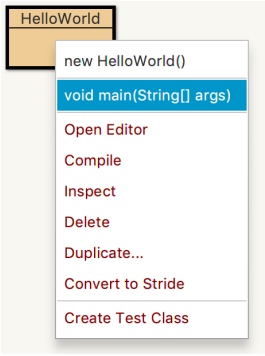
6. Type the code to the right EXACTLY as it appears on this page, paying attention spacing and each character. Use your tab key to ensure lines 2 - 4 are correctly spaced. A single misplaced character can cause a program to be unusable.



7. Take a moment to organize your windows. You'll need to be able to see your workspace and editor windows at the same time.

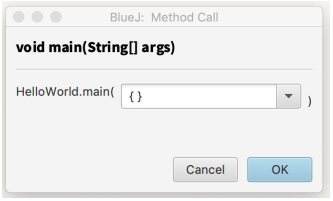


8. Click Compile. This is somewhat like Java's "spell-checker". If you've made an error in your code, the slashes will turn red and the bad line of code will become marked in the editor. Make corrections and attempt compiling again until the slashes clear. You are now ready to run your program.

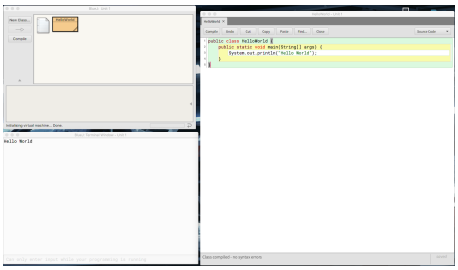


9. Right click on the HelloWorld box in the class workspace. Click 'void main(String[] args)'.

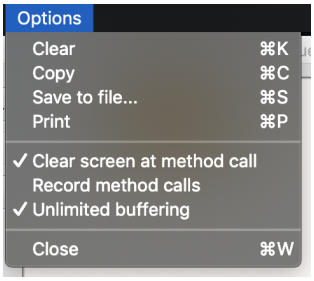
10. A box will appear asking if we want to provide any additional information before executing this program (we never will). Click OK.



11. You should see a third window open. We call this window the console. In it you should see the words Hello World. Congratulations! You just wrote your first program!



12. Take a moment to reconfigure your three windows to match the image on the right. We always want to have as much vertical space as possible to see our code, but we also need to be able to see the workspace and the console. Students who don't organize their windows will waste a lot of time manually switching between them



13. Open the Options menu on the console window. Turn on 'Clear screen at method call' and 'Unlimited buffering'.

Reflection Question

How could you change your code to get your name to appear on the screen? *(Try it out and then write your answer in the space below).*

Task 2: Print Statements

One of the most important things a program can do is display information in the console. This exercise will help you familiarize yourself with the fundamentals of print statements. *(Answer the numbered question on this paper)*

1. Execute the program seen at the right. What appears in the console?

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
        System.out.println("Hello World");
        System.out.println("Hello World");
    }
}
```

2. Change your program so that the characters printed in the console look like a smiley face.
   a. How may print statements did you need?
   b. What characters did you use?
   c. Were there any characters that caused the compiler to fail to compile?

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

3. Sammy writes the code on the right to try to get a quotation mark to print to the console, but her code fails to compile. Why?

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

4. Try each of the following print statement in your code. What appears in the console? Note the times when two characters don't appear as two characters in the console.
   a. System.out.println("\"");
   b. System.out.println("\tt");
   c. System.out.println("\tx");
   d. System.out.println("\nn");
   e. System.out.println("\nm");
   f. System.out.println("\\");
   g. System.out.println("\\\\");
   h. System.out.println("\");

5. The combination of two characters can make an *escape sequence*. Programmers make use of them to help print statements. Guess the purpose of each escape sequence below:

   \" \\ \t \n

6. Java provides two types of print commands: println and print. Try using a combination of the statements below to see how each behaves. What is the difference between println and print?
   a. System.out.println("a");
   b. System.out.println("");
   c. System.out.print("b");
   d. System.out.print("");

Lecture Notes: Hello World!

Vocabulary

IDE	
escape sequence	
print statement	
console	
compile	

Reflection Questions

1. What is the Lynel Rule?

2. What edition of Building Java Programs do we use on PracticeIt?

3. What is the correct configuration of the three BlueJ windows?