1.05 Virtual Lecture Notes

Part 1

If this is the first time you have tried to create a project or class with BlueJ, you may want to refer to the BlueJ Quick Start Guide you downloaded in the last lesson.

1. Create a new project called 1.05 HelloWorld in the Mod01 Lesson folder.
2. Create a new class called HelloWorld in the newly created project. Remember that class names in Java do not include spaces, and the first letter of any word is capitalized. BlueJ will use the name of the class as the file name when your program is saved. The name of the file and the class must match.
3. Open the HelloWorld class. When BlueJ creates a new class, it provides a template for the design of a class. Delete everything between the opening and closing curly braces before you type any code.
4. Type in the source code as shown in the 1.05 Desk Check document, but remember: Don't type the line numbers.
5. Compile the program. If you typed the code exactly as shown, it should be error-free. If you do get error messages, carefully compare the original source code to what you typed and fix any errors.
6. Run the program. When you run the program a message should appear on the screen.

Reflect on your reaction to making the computer follow your instructions for a change. You may even want to share your accomplishment with a family member!

Part 2

Now let's make some modifications to the HelloWorld class.

- Place your cursor after the semicolon at the end of Line <9> and press the Enter key to insert a blank line.
- Type the new highlighted statement shown below.

```
/**
 * Display a "Hello World!" message on the screen
 *
 */
 public class HelloWorld
 {
    public static void main(String[] args)
    {
           System.out.println("Hello, Virtual World!");
           System.out.println("It is a great day for programming.");
    }//end of main method
}//end of class
```

Compile and run the program with this new statement. If it does not execute correctly, check the line and fix any mistakes. If the program ran successfully, try adding additional lines to print other messages. What happens if you put any empty print statement, like the following, on a line between any two print statements?

```
System.out.println();
```

Print statements will be an important part of every program because they allow you to display information on the screen.

Part 3

The quickest way to learn anything, including programming, is through play and practice … so now let's mess things up! Make the modifications listed below, one at a time, then compile and run the program. Record your observations about what happens, you may need them for one of the upcoming assessments. If an error occurs, restore the change you made to its default setting before making the next change.

1. Replace the forward slash (/) on line <1> with a backward slash (\).
2. Remove the last asterisk (*) on line <1>.
3. Remove the asterisk on line <2>.
4. Delete the forward slash on line <3>.
5. Delete public on line <7>. Capitalize the p in public.
6. Delete class on line <7>. Capitalize the c in class.
7. Add a space between Hello and World on line <5>.
8. Change the case of any letters in the class name on line <5>.
9. Change the name of the class on line <5> to anything else.
10. Delete any of the curly braces.
11. Change the case of System, out, or println in line <9>.
12. Delete "ln" from the end of println on line <9>. (Note: this will be important in the upcoming assignment!)
13. Remove the dots in line <9>.
14. Remove the parentheses in line <9>.
15. Remove the quotation marks in line <9>.
16. Delete the semicolon at the end of line <9>.
17. Type something different between the quotation marks on line <9>.
18. Type the value of pi between the quotation marks on line <9>.
19. Type the value of pi inside the parentheses on line <9> without quotation marks.
20. Add a backslash (\) at various points inside a pair of quotation marks.

If you made all of these changes and fixed any errors that resulted, you have gained some valuable insight into how precisely Java programs must be written. Remember, Syntax Rules!