## Lesson 1.2 “Hello, World!” Writing a First Java Program

### Writing the program in a text editor

```
/**
  * HelloWorld.java program
  * @author <your name>
  * @version <the date>
*/

public class HelloWorld
{
    public static void main(String[] args)
    {
        // Display a greeting to the user
        System.out.println("Hello, World!");
    }
}
```

Once you’ve written the program, save it as **HelloWorld.java**

———————————————————————-

Some initial notes about Java:

-Java is case-sensitive.

-Java has free-form layout. But we will use strict layouts.

-The filename of the program on your computer, **HelloWorld.java**, has to be the exact same name as the **public** class in the program, **HelloWorld**.

Some specific notes about this program:

- **comments:** The five lines at the beginning of the program, beginning with **`/**`** ... These are *comments*, lines that are of absolutely no use to the compiler—it ignores them when creating the bytecode for this program—but of enormous benefit to the humans who are writing, reading, and editing this code.

    - s**ingle-line comments** begin with a **`//`**,
    - **multi-line comments** are enclosed in **`/*`** and **`*/`**,
    - **`/**`** indicates that these comments are available to users in a JavaDoc format.

- **`public class HelloWorld`**
  ... starts a new class. In Java, every source file (a file that ends with the extension **`.java`**) can contain only one public class.

- **`public static void main(String[] args)`**
  ... Defines a method called **`main`**. Every Java application has to have a **`main`** method.

- **`// Display a greeting to the user`**
  ... is an explanatory *comment*, and thus is ignored by the compiler.

- **`{`** and **`}`**
  ... these *curly braces* enclose a series of instructions that are executed one by one.

- **`System.out.println("Hello, World!");`**
  This statement displays text on the screen. Let's analyze this statement:

  - **`System`** is a *class* that contains objects and methods that supply access to system resources (like your monitor);

  - **`out`** is an *object* that has various methods we can use to output information;

  - **`println`** is a *method* that tells **`System.out`** what to do with the text.

  - Note that this statement, as do all instructions (that don't have curly braces) ends with a semicolon.

- Whenever you call a *method* to do something in Java, you have to include three things:

  1. The object you want to use (**`System.out`** here)

  2. The method you're going to be using for that object (**`println`**)

  3. Information that the method needs access to, enclosed in parentheses. If the method doesn't need any information, then you just use an empty set of parentheses: **`()`**.

## Practice

### Exercise 1

Before you do anything else, find out how to compile and run a Java program. Some environments provide sample programs similar to the example in Section 1.3.

1. Type in the Hello World program; then compile and run it.
2. Add a print statement that displays a second message after the `Hello, World!`. Say something witty like, `How are you?`. Compile and run the program again.
3. Add a comment to the program (anywhere), recompile, and run it again. The new comment should not affect the result.

This exercise may seem trivial, but it is the starting place for many of the programs we will work with. To debug with confidence, you will need to have confidence in your programming environment.

In some environments, it is easy to lose track of which program is executing. You might find yourself trying to debug one program while you are accidentally running another. Adding (and changing) print statements is a simple way to be sure that the program you are looking at is the program you are running.

### Exercise 2

It is a good idea to commit as many errors as you can think of, so that you see what error messages the compiler produces. Sometimes the compiler tells you exactly what is wrong, and all you have to do is fix it. But sometimes the error messages are misleading. Over time you will develop a sense for when you can trust the compiler and when you have to figure things out yourself.

Starting with the Hello World program, try out each of the following errors. After you make each change, compile the program, read the error message (if there is one), and then fix the error.

1. Remove one of the opening curly braces.
2. Remove one of the closing curly braces.
3. Instead of `main`, write `mian`.
4. Remove the word `static`.
5. Remove the word `public`.
6. Remove the word `System`.
7. Replace `println` with `Println`.
8. Replace `println` with `print`.
9. Delete one parenthesis.
10. Add an extra parenthesis.

Read. Chap 1.  Think Java
https://greenteapress.com/thinkjava7/html/chapter-01.html#sec11