

Homework #1 (due on 9/10/24)

This is a warm-up assignment to make sure every student has a working set-up and can write a small java program. In class we will be using eclipse and the jdk (java development kit), both freely available. If you do not have your own laptop, and will be using the ITS labs, they should already have this software installed, so you can skip the next section.

Java programming using the command line – Mac OS (50%)

To manage, compile and run your java programs from the console you will need to use three items: a terminal window (*Terminal*), a text editor (like *TextEdit*), and a window showing your home directory, as well as some java commands (**javac** and **java**).

- 1) Dock a *Terminal* window: go to *Finder* → *Go* → *Utilities* → *Terminal* and drag it to the dock.
- 2) Dock a *TextEdit*: go to *Finder* → *Go* → *Applications* → *TextEdit* and drag it to the dock.
- 3) You need to change the following keyboard and *TextEdit* configurations:
 - a. Go to *Apple* → *System Preferences* → *Keyboard* → *Text* uncheck “Use smart quotes and dashes”
 - b. Open a *TextEdit* window (it should be in the dock). Go to *TextEdit* → *Preferences* → *New Document* and change the *Format* to “Plain Text” and under *Options*, uncheck: “Smart quotes”, “Smart dashes” and “Smart links”
 - c. Go to *TextEdit* → *Preferences* → *Open and Save* and change “Opening files” and “Saving files” to “Unicode UTF-8”
- 4) You should also open a *Home* window to look at your files: *Finder* → *Go* → *Home*
- 5) Make sure that java is running in your computer:
 - a. Open a *Terminal* window
 - b. Type **java -version**, and you should see some output with the version of JRE (Java Running Environment) that your computer is running.
 - c. Type **javac -version**. If you see a version number you are ready to go; if you don't then you need to install java JDK (Java Developers Kit) from Oracle (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)
 - d. After the installation, when you type **javac -version**, you should see a version number
- 6) Writing your first Java program:
 - a. Open the *TextEdit* application.
 - b. Type the following code:

```

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

```

- 7) Saving your program: Under *TextEdit* → *File* → *Save*, type **HelloWorld.java** next to *Save As*, select *Where* you want to save the file and then click on the *Save* button. You should see the java file in the *Terminal* window when you type **ls** followed by a return (enter). If you don't, explore in the *Home* window, where is the java file located. Once you locate the file, you need to change the directory in the *Terminal* window to the directory where the java file is located.
- 8) Compiling your program: type **javac HelloWorld.java**. If you now type **ls**, you should see to files: **HelloWorld.java** (the source code, i.e. what you typed in *TextEdit*) and **HelloWorld.class** (the bytecode that will be interpreted by the Java Virtual Machine).
- 9) Running your program: type **java HelloWorld** and you should see the output of the program: **Hello World**.

'Song Lyrics' programming assignment (50%)

Write a program that prints out the lyrics to one or several verses of your favorite song (you can use java statements like *System.out.println("....");*). Be sure to include comments in the program indicating the title of the song and the author. Pay attention to formatting using the following codes:

```

\n line break
\t tab
\" double quote
\' single quote
\\ backslash

```

Experiment with the different special characters to format your lyrics. There is a reference for some of these special characters in the book on page 126. Some basics of using eclipse to write your first program are at the end of chapter 1.

Grading

Does the program compile? If not, you will lose all the points for that problem.
Is the program properly documented? (worth ~50% of the problem)

Proper documentation includes:

- a preamble with the name of the author, date of creation and brief description of the program.
- comments inside the code describing steps needed to be taken to accomplish the goal of the program.
- appropriate formatting, indentation and use of white space to make the code readable.

Remember that the code is read by humans and it should be easy to read for people who were not involved in its development.

Is the program correct? (worth ~50% of the problem), Make sure that your program produces results as specified above.

What and how to submit?

Homework should be submitted through *NYU Classes*. It should be submitted by the due time and date, or it will be marked as late. To submit homework, upload a single file (just the source code, i.e. files ending with the .java extension, not the .class extension). If there is more than one file, upload a single zip file please.