CS102A Introduction to Computer Programming
Fall 2020
Lab 1

Objectives

1. Install JDK and configure the Path environment variable.
2. Compile and execute your first Java program using the command line.

1 Course Logistics

1.1 Lab Exercises

Each lab will include a number of programming exercises designed to consolidate the knowledge gained from the lectures. You should attempt to solve these exercises during the lab. You are encouraged to approach the lab instructor & assistants for help.

1.2 Assignments

There will be four assignments throughout the course of the semester. These assignments are independent of the lab exercises and will be sent to you via the Online Judge system. You should submit each assignment before the set deadline.

1.3 Project

Towards the end of the semester, all students will form groups of two and collaborate on a Java project.

2 Software Installation

2.1 Installing JDK

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes a Java Runtime Environment (JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc), and other tools needed in Java development.

You may download and install JDK from the following link: `https://www.oracle.com/java/technologies/javase-downloads.html`. Once the executable has been downloaded, run it and follow the prompts to install JDK. **Do not change the default installation path**.

> **Note**
> You may also download JDK from Sakai.

Today's exercise is to make sure that JDK is correctly installed on your machine. Open a terminal window or command prompt, then type:

```
java -version
```

If running the above command returns the following message:

```
java version "1.8.0_261"
Java(TM) SE Runtime Environment (build 1.8.0_261-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)
```

then you have successfully installed JDK on your machine.

> **What is the command line?**
> The command line is a text interface for your computer. It is a program that accepts user commands and forwards them to the computer's operating system to be executed. Using the command line, you can navigate files and folders on your computer, just as you would with Windows Explorer on Windows or Finder on Mac OS; the difference is that the command line is fully text-based.
>
> - On Windows, press the `Win` and `R` keys together. In the prompted window, type `cmd` and press `Enter`.

- On MacOS, locate the *Terminal* application.

What are JVM, JRE, and JDK?
`https://www.geeksforgeeks.org/differences-jdk-jre-jvm/`

2.2 Java Environment Variables

If the `java -version` command returns the following message:

```
'java' is not recognized as an internal or external command,
operable program or batch file.
```

it means that your machine cannot find JDK. In this case, you need to manually set up the Path environment variable.

What are environment variables?
An environment variable is a dynamic-named value that can affect the way running programs will behave on a computer. For instance, environment variables help programs know where to install files, where to store temporary files, and where to find user profile settings. The Path environment variable is used by an operating system to locate needed commands (e.g., `java`) within a list of directory paths.

1. Press the `Win` and `R` keys together. In the prompted window, type `SystemPropertiesAdvanced` and press `Enter`.
2. Click on the *Environment Variables...* button on the bottom right corner.
3. Double click the *Path* line in *System variables* groupbox.
4. Click on the *New* button, then input the following string and press `Enter`:

   `C:\Program Files\Common Files\Oracle\Java\javapath`
5. Click *OK* on all windows when prompted.

You will then need to close and re-open your command line prompt to verify whether `java` is now accessible.

2.3 Editor

Any text editor can be used to program in Java. In this lab, we will be using IntelliJ IDEA. You may download and install it from: `https://www.jetbrains.com/idea/download/`.

> **Note**
> A popular alternative is Visual Studio Code, which you may find at: `https://code.visualstudio.com/`.

3 Exercise

3.1 Compile a Java Program Using the Command Line

After creating a new project folder on your machine, open your text editor and create a new text file with the following code:

```
public class Demo {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Use the *Save As* command in the *File* menu to save the file with the name Demo.java in the created project folder. Note that the .java file name has to be the same as the main class name in your code.

> **Note**
> If you are using Visual Studio Code and a *Java Extension Guide* pops up, click the *Install All* button.

Next, open the command line and use the `cd` command to go to the directory where you have saved your Demo.java file. Then, use the `javac` command to compile your .java file. Finally, use the `java` command to execute the .class file (which must have a main function). The procedure is as follows (the `>` sign means that the corresponding line is a command to be executed):

```
C:\Users\James>cd "Desktop\lab 1"
```

```
C:\Users\James\Desktop\lab 1>javac Demo.java

C:\Users\James\Desktop\lab 1>java Demo
Hello, world!

C:\Users\James\Desktop\lab 1>
```

Note

When executing the compiled .class file, the .class extension must be removed from the command. Otherwise, the following error will appear:

```
C:\Users\James\Desktop\lab 1>java Demo.class
Error: Could not find or load main class Demo.class
Caused by: java.lang.ClassNotFoundException: Demo.class
```

Note

If you are using Visual Studio Code, you do not need to manually compile and run the file from command line. Instead, you can do both at once by selecting *Run* from the top menu.