Exercises module 1

Exercise 1 – Compile java from the command line

The purpose of this exercise is to compile and execute a simple "**hello world**" program from the command line without any IDE (Integrated Development Environment).

These exercises assumes you have the JDK standard edition (SE) installed from oracle.com

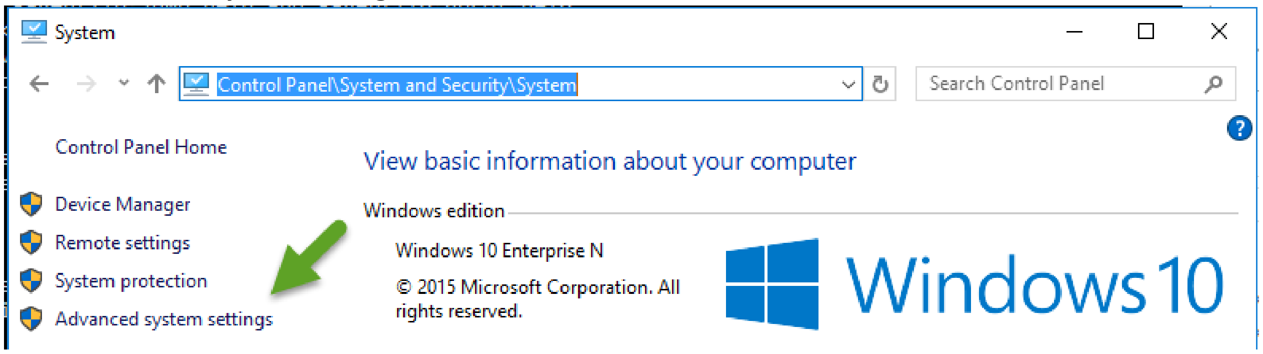
1. Use Notepad++ (https://notepad-plus-plus.org) to create a new text file named **hello.java** in this folder: **c:\temp**
2. Write the following code in the file and then save it as **hello.java**:

```
public class Main{

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

}
```

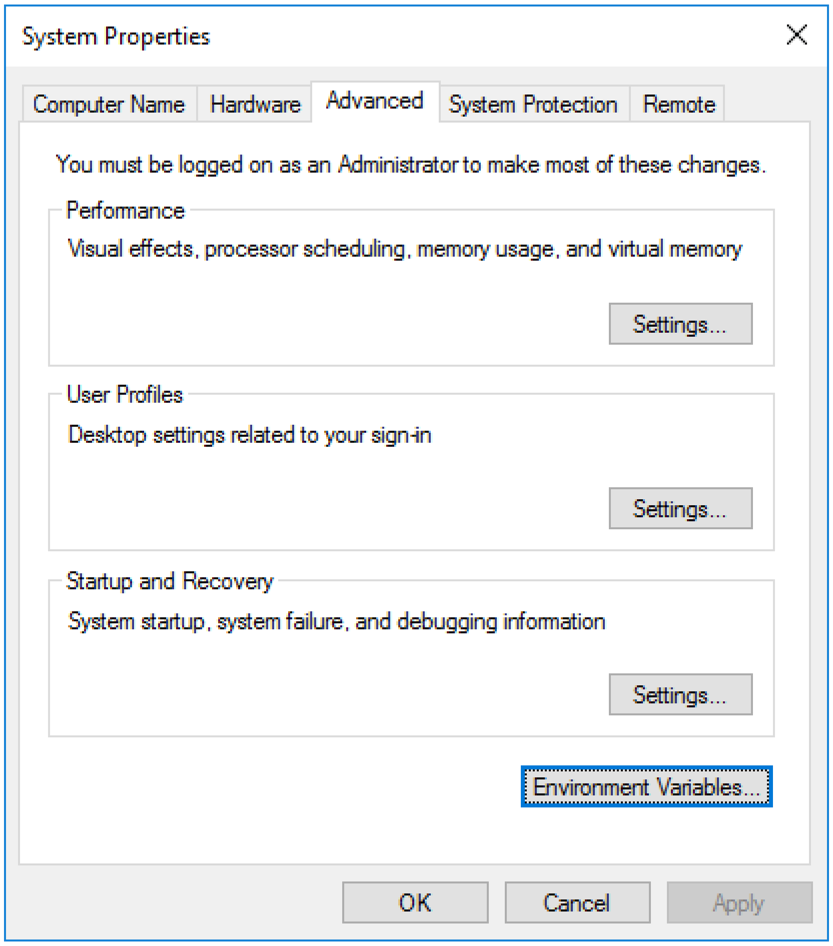
3. Now we need to compile this file to create a **hello.class** file using the **Javac** tool.
4. Open a new command prompt and run **javac**. If you get this error then do the following steps, if not jump to step 11.

```
C:\temp>javac
'javac' is not recognized as an internal or external command,
operable program or batch file.
```

5. Open the control panel and go to the **Control Panel\System and Security\System** page and click on the **advanced system settings**



6. **Under the Advanced** tab click on the button "Environment Variables…" at the bottom of the window:



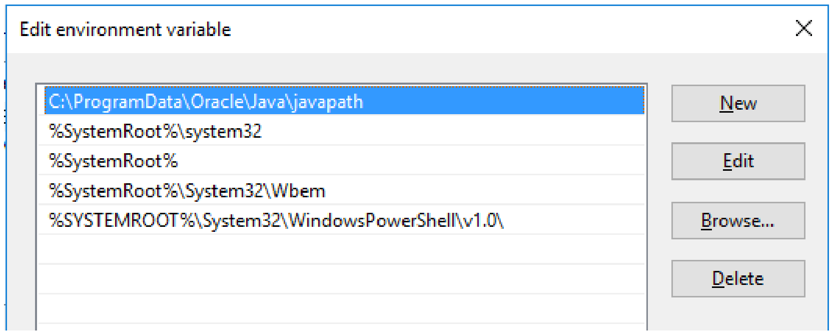
You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

Startup and Recovery
System startup, system failure, and debugging information

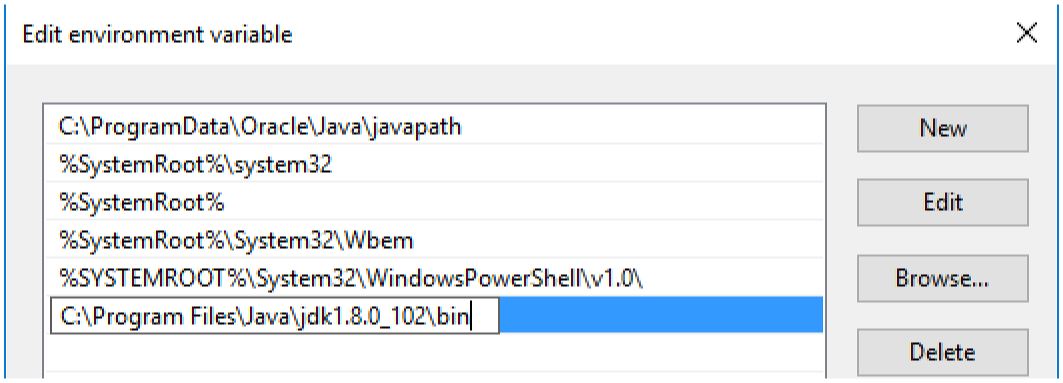
7. Look for the Path system variable and double-click on it. A new window should open looking like this:



```
C:\ProgramData\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
```

8. We now need to locate the full path to the java development kit (JDK). Open a new explorer window and navigate to one of these folders:
C:\Program Files\Java\jdk1.8.0_102\bin
C:\Program Files (x86)\Java\jdk1.8.0_102\bin
(The exact numbers is not important here).

9. Copy the path from the explorer window into a **New** entry in the environment variable list, like:



```
C:\ProgramData\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
C:\Program Files\Java\jdk1.8.0_102\bin
```

10. Now **close** any command prompt windows and open a new command prompt window. Type **Path** and you should see the newly added path in the result.

11. If you are not familiar with **path** and the **environment variables**, then read up on that before you continue. For example, this is a good starting point: http://superuser.com/questions/284342

12. To test that everything works run **javac** from the command prompt and you should see something similar:

```


C:\Users\tore>javac
Usage: javac <options> <source files>
where possible options include:
  -g                          Generate all debugging info
  -g:none                     Generate no debugging info
 . . . .
```

(Now everything is setup correctly)

13. **Javac** is the primary Java compiler that is part of the JDK and we use it to compile our java code.

    Now go to **c:\temp** in the command prompt and now let's compile **hello.java** that we created earlier. Lets run **javac hello.java**

14. Unfortunately, we get an error:

```
C:\Temp>javac hello.java
hello.java:1: error: class Main is public, should be declared in
a file named Main.java
public class Main{
       ^
1 error
```

The reason for this is the various rules for **.java** files:

- Each source file can only contain one public class
- The file name and the class name must match
- The capitalization (upper/lower-case) should also match between the file name and the class name. Even though some operating system does not care about this.

15. Rename the file to **Main.java** and recompile it using **javac Main.java** and this time it should work.

16. Now do a **dir** to display the content of the **c:\temp** folder and you should now see the compiled result in a new file named **Main.class.**

```
C:\Temp>dir
 Volume in drive C has no label.
 Volume Serial Number is 6F6F-14E6

 Directory of C:\Temp

07/30/2016  02:46 PM    <DIR>          .
07/30/2016  02:46 PM    <DIR>          ..
07/30/2016  02:46 PM               414 Main.class
07/30/2016  02:45 PM               123 main.java
```

17. Try to open **Main.class** in **Notepad++.** As you see the result is a binary file that is not editable or viewable in Notepad++. This is because this is a binary file that is not human readable.

18. Now we want to execute our application **Hello World!** Try to run **main.class** or just **main**. But you notice that it does not work! Why? The reason I that your operating system does not know how to execute a **.class** file directly.

    Read https://en.wikipedia.org/wiki/Java_bytecode to get more information about the Java bytecode.

19. To execute compiled java byte-code we need to execute them inside the Java Virtual Machine (JVM). To do this we run **java Main** and you should see this output:

```
C:\Temp>java Main
Hello, World

C:\Temp>
```

    What happens if you try to run “java main” (all lower-case characters?)

20. To see the current version of the JVM used you can run **java -version** and you should get the following result:

```
C:\Temp>java -version
java version "1.8.0_102"
Java(TM) SE Runtime Environment (build 1.8.0_102-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.102-b14, mixed mode)
```

21. Read about **Java Hotspot** on Wikipedia here: https://en.wikipedia.org/wiki/HotSpot

Questions and concepts to study further on your own:

- Difference between **Javac** and the **java** command-line tool
- The difference between Java **JRE** and **JDK**
- Environment variables
  https://en.wikipedia.org/wiki/Environment_variable
- The path variable
  https://en.wikipedia.org/wiki/PATH_(variable)
- Java naming conventions
- Java Just-In-Time compilation (**JIT**)
- Java bytecode
- Virtual machine
- Interpreter vs compiler
- Java vs C/++ vs C#
- Who is James Gosling