Extra Lab Sheet 1:
Introduction to Java programming

This section will help you getting started with Java programming using the machines in the ITL. You will need to know your User ID / Password of your EECS Account. This is different to your college one and can be found on the pink sheet that you were given. The following instructions are written as thought you are using the machines in the ITL under Linux. If you are not using the lab facilities, some of the details below might be slightly different, based on the operating system you are using.

For any enquiries, feel free to use the QM+ Discussion Forum outside of the lab hours.

Getting started

1. **Log in under Linux** using your EECS Account described above. If the machine is currently running under Microsoft Windows, you will need to restart it and choose Linux from the available option on the start-up menu.
2. **Open a desktop terminal window**, by clicking on *Activities* (top left corner) and *Terminal*.

   The terminal is a tool for input and output operations for a Linux system. You can write commands and by pressing 'Enter' they will be executed. You can use them to start programs, create folders, compile Java programs etc. This is a short list of the most common commands you will need throughout this course:

`pwd`	Print the path of the current working directory. (**p**rint **w**orking **d**irectory)
`ls`	**Li**s**t** the files of a directory
`mkdir <name>`	Create a new directory (**m**a**k**e **dir**ectory)
`cd <directory>`	Change the current working directory (**c**hange **d**irectory) (`..` for parent directory)
`gedit &`	Load gedit text editor
`javac <name>.java`	Compile the given java source code
`java <name>`	Run the compiled java program

   You can find even more commands in this documentation:

   http://linuxcommand.org/learning_the_shell.php

3. **Create a folder structure.**
   a. Type "`pwd`" in the terminal window and press Enter. This command will show you your current directory. Every command you perform will be performed in this directory (e.g. folders will be created here).
   b. Create a new directory named `ecs717` to work at. You will save the lab's work into this directory. You can do this by typing "`mkdir ecs717`" on the terminal window.

c. Let's inspect the directory content with the "`ls`" command. You should see the new `ecs717` folder.
d. Move into the new directory using the command "`cd ecs717`". Every command you perform now, will be performed in this folder.
e. Create another directory (inside `ecs717`) to use for today's lab work (e.g. `lab01`). Move into this directory. (You can always go back to the parent directory using the command "`cd ..`")
f. Use the command "`pwd`" to verify that you are inside the correct directory (the result should look similar to "`/home/user/ecs717/lab01`").

4. **Write your first Java program** using the text editor *gedit*. Use the command "`gedit &`" to open it. Type the following code:

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

Save the program as `Hello.java` on the folder you previously created (e.g. `ecs717/lab01/Hello.java`). In the terminal window, use the command "`ls`". It should list all the files inside the current directory (`Hello.java` in this case).

5. **Compile your program** using the command "`javac Hello.java`". This process transforms your human-readable instructions in the file `Hello.java` to machine readable bytecode in a file `Hello.class`.
Use the "`ls`" command again to verify that the file `Hello.class` has been successfully created.

6. **Run your program** using the command "`java Hello`" (You do not put the file ending .class here).
The message "`Hello World!`" should appear on the terminal window.