

Computer lab – Week 2

Welcome to the computer labs!

- You will receive instructions in the beginning of each computer lab.
- Try to do all exercises. If you can't finish during the computer labs, finish later.
- Ask questions!!! A lot of them!! Check in Canvas for different ways to get help and ask more questions.

1. Login (if you are using a computer lab at UoN)

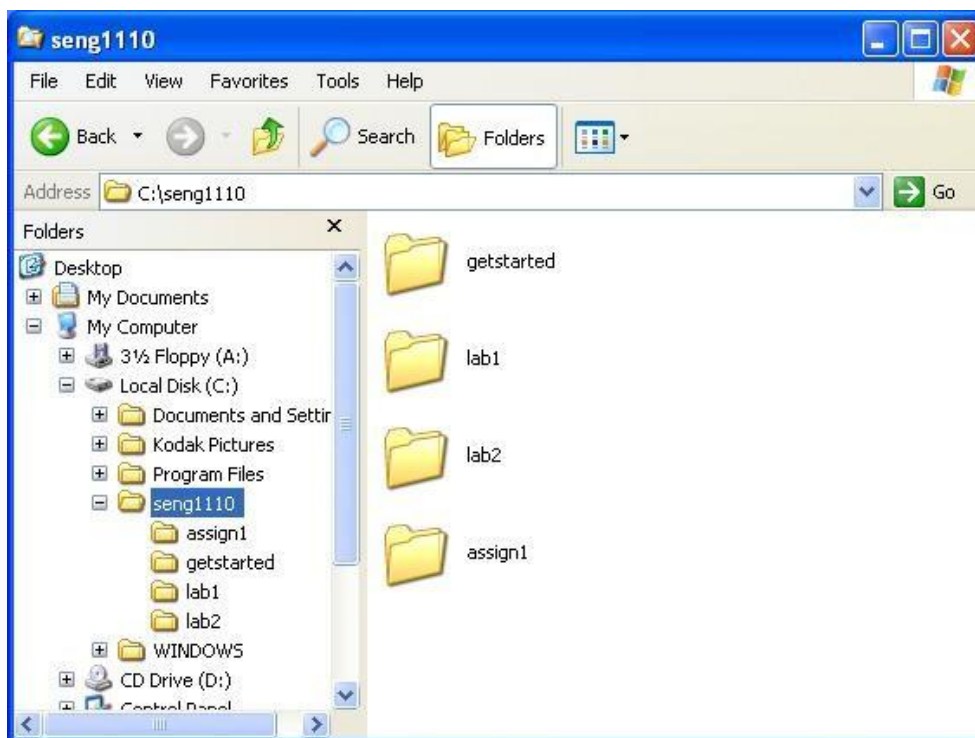
- a. Using the student number (remember that you must include the lower-case 'c', for example c7020012) and initial password provided on your student card, together with the instructions given to you on the handout, log on to your terminal.
- b. Change your initial password to one that you can remember as instructed by your lab demonstrator.

2. Web pages

- a. Login in Canvas and go to SEN6110 course. Click on the links to the lecture slides, computer labs, etc, but DO NOT DOWNLOAD A LOT OF FILES because these will waste your time at the moment and unnecessarily consume your disk storage quota.

3. Preparing a subdirectory to working

- a. Create a folder called “seng1110 (or seng6110)” in your U: drive (if you are in a computer lab at UoN). You should keep all your .java files for SENG1110/SENG6110 in here. Keep them organised by creating more folders inside this folder as you need to.
- b. As you can see in the picture below, I have created a couple of folders for some labs, and an assignment.



- c. Make a subdirectory called “lab1”.

- d. In Canvas go to Week 2 – Java basics – part 2 link and computer lab link.
- e. Open the instructions file.
- f. Save the file `Salary.java` in the directory `lab1`. Use the mouse right button to save.

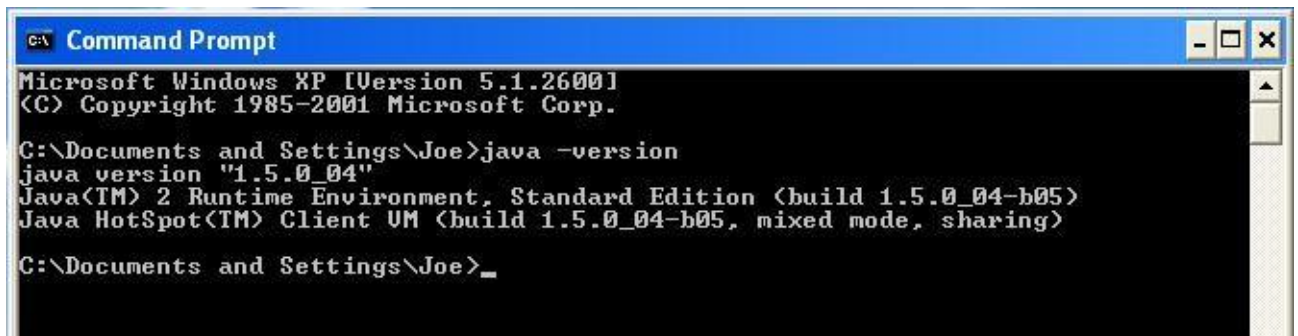
#### 4. Editing

- a. A source file is a `.java` file. Source files contain Java code that you write to specify what your program will do. The program we will write now consists of only one source file. To edit it, open it with Notepad++.
- b. You don't have to use Notepad to write and edit your `.java` files. You can use any text editor, as long as you save your files as plain text. There are lots of good editors out there that are well suited to editing Java files; some are free and some are not.

If you find a good editor tell others about it on the discussion board.

#### 5. Open a command prompt window.

- a. From the Start menu, open the Command Prompt. If you can't find it, click Start, Run, and type "cmd".
- b. A window will appear with a prompt. Test to make sure you have installed Java correctly by typing "java -version". You should see a message like this:



```
C:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Joe>java -version
java version "1.5.0_04"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_04-b05)
Java HotSpot(TM) Client VM (build 1.5.0_04-b05, mixed mode, sharing)

C:\Documents and Settings\Joe>_
```

- c. If you get the message "**java is not recognised as an internal or external command**" and you are at home, make sure that you have followed all the instructions located in week1/computer lab in Canvas. If you can't get it working, you can ask your demonstrator during the computer lab and/or post your question in the discussion board.

#### 6. Compile your source file to produce a bytecode file.

- a. The command prompt shows your current directory. In Windows XP, the command prompt starts in your user directory, in my case `C:\Documents and Settings\cxxxxxx`. You may want to change your current directory to `U:\seng1110\lab1` so that you can work with files in there. You can do that with the command

`"cd /d. u:\seng1110\lab1".`

Note that in your computer lab you will be working in the `u:\` and at home you will be working in a different folder.

Once you're in the right directory, you should be able to compile your program. Use the command "javac `Salary.java`". If no messages come up, it means the program compiled successfully. If you are lucky enough for that to happen, check the contents of your folder. It should now have a second file in it called "`Salary.class`". This is called a Java bytecode file.

If you do get some error messages, make sure you have downloaded the Java code in from step 3 f) exactly as it appears, with correct capitalisation.

```
C:\SENG1110\lab1>javac Salary.java
C:\SENG1110\lab1>
```

7. Run your program.

Now that you have compiled your program, you can run it. While still in the “lab1” directory in the command prompt, use the command “java Salary”. Your program should work its magic, displaying a satisfying message and then terminating. If it does that, congratulations!

```
C:\SENG1110\lab1>javac Salary.java
C:\SENG1110\lab1>java -version
java version "1.7.0_02"
Java(TM) SE Runtime Environment (build 1.7.0_02-b13)
Java HotSpot(TM) 64-Bit Server VM (build 22.0-b10, mixed mode)
C:\SENG1110\lab1>javac Salary.java
C:\SENG1110\lab1>java Salary
Please Enter number of normal hours:
```

**Quick-reference for the command prompt**

cd c:\sengl1110\getstarted	move you to the named directory
cd myfiles	move you into the directory “myfiles”
cd ..	move you up one directory
cd \	move you to the top directory
d:	move you to D: drive
dir	display files in the current directory
javac Welcome.java	compile the “Welcome.java” source file
javac *.java	compile all source files in the current directory
java Welcome	run the program called “Welcome”

8. Make other changes to the program code:

- Suppose that all workers receive at least a salary of 200 dollars, independently of the number of hours worked. How to add this new aspect in the program?
- Change the messages
- Add a header comment following the example below:

```
/*Salary.java - Lab 1
 *Author: Regina Berretta
 *Modified by: your name
 *Student No: 3XXXXXXx
 *Date: dd-mm-yyyy
 *Description:
 *Program to calculate weekly salary from total number of hours worked
 *Modified to include minimum retainer pay of 200 dollars per week.
 */
```

- Add comments using // or /\* \*/ in the code explaining each line.

9. Watch video Lab-week2b-Hello\_World. It shows the basics to write your own code.

10. Open an editor and write the code below. Save the file as Average3.java. You will find the video Lab-week2cImplementing\_Addition in Canvas that can help you to complete this task.

```

public class Average3
{
    public static void main(String[] args)
    {
        int num1 = 75;
        int num2 = 137;

        double average = (num1 + num2)/2;

        System.out.print("The average of " + num1);
        System.out.println(" and " + num2 + " is " + average);
    }
}

```

**Figure 1.**

Compile and run "Average3.java." and observe the output. The program declares three variables; two variables hold integer numbers that have been provided within the program, and one variable of type double is declared to store the result of the calculation. The program then displays the average of the two numbers.

Modify the program as follows:

1. Create an integer variable named **num3**.
2. Assign the value of 265 to **num3**.
3. Modify the average calculation to include **num3**.
4. Modify the output statements as appropriate.
5. Compile and run the program. If you encounter any errors (syntax or logic) that you cannot resolve, ask the lab instructor or assistant for help.

11. Write a code for the SMS message cost problem that we have discussed during the lecture:

- i. Input number of Messages (it will be a whole number)
- ii. Unit Cost of each Message (say 22 cents per message)
- iii. Setup cost = 10 dollars
- iv. Output total cost

12. Write a pseudocode and a Java code that calculates the salary of an employee per week in 2021. Suppose the user will inform his/her name (first name + last name), his/her salary in 2020 and the salary increment for 2021 (percentage). Then, calculate the new salary and divide the total for 52 weeks. Output the name of the employee and the salary per week