Worksheet Two

"Don't Panic"
Douglas Adams, The Hitch Hikers Guide to the Galaxy 1979

Unit Learning Outcomes Addressed by this worksheet: 1 & 2

Prior to coming to this practical you must have:
- Read through and summarised modules one and two from the lecture notes.
- Read through the how to print from Unix handout in the St151 unit web area.
- Double checked via the unit web area that you are attending the correct practical.

All of the Java code for this worksheet should be placed in the P02 directory that you created lasted week.

Exercise One (Developing ground truth data)
The formula below is for converting a temperature from Fahrenheit to Celcius.

$$Celcius = \frac{(Fahrenheit - 32.0) \times 5.0}{9.0}$$

Using a calculator, generate 5 Fahrenheit/Celcius pairs by hand. These values will be used to test your solution to exercise four.

Exercise Two(Algorithm design)
Design, in pseudo code, an algorithm which converts degrees Fahrenheit to degrees Celsius. An amount in Fahrenheit is input, converted to Celcius and then output to the user.

Exercise Three (practice with Java implementation).
Translate your algorithm into Java and then compile and test it. It is probable that your program will not compile. Examine any compiler error messages carefully. Keep correcting any errors until you successfully manage to compile your Java application. Run it using the Fahrenheit/Celcius pairs that you generated for exercise two. Does you Java application generate the same answer? If it does not then seek guidance from your tutor.

Exercise Four (practice with using sub modules).
Finish the algorithm below by designing the sub modules and adding the IMPORT/EXPORT information to the sub module calls in the main algorithm below. The algorithm inputs a measurement in pounds and converts it to a measurement in kilograms. Please note that the MAJOR point in this exercise is to gain some familiarity with the use of sub modules. Use the inches to centimetres example from the module two lecture notes to guide you.

```
MAIN
   INPUT pounds
   convertPoundsToKilos
   outputPoundsAndKilos
END
```

Exercise Five (practice with using sub modules in Java).
Convert your pseudo code from exercise four into Java. Place it in a file called PoundsAndKilos.java. What does the name of the Java file imply in terms of the name of the class inside the file?

Exercise Six (making backups).
It is always a good idea to keep a backup of your ST151/501 work on a usb drive. If you don't have one the I recommend buying one (it doesn't need to be a very big one for ST151/501 backups). All of the lab computers have usb ports. From the Linux labs:

- Insert your usb drive into a usb port
- After a short while the disk icon for the USB drive will appear on the desktop.
- The usb drive will be mounted on /media and will appear as a directory with the same name as you have given your usb drive.
- You can copy files to/from the usb drive in the usual manner.
- When you are finished, right click on the drive icon and select unmount device from the menu.
- You can now remove your usb drive.

Exercise Seven (printing files).
Use the how to print from Unix handout (obtained from the unit web area) to print out all of your .java files so far.

Exercise Eight (Another algorithm with sub modules)
Electric powered radio controlled model aircraft these days make use of Lithium Polymer (Lipo) batteries for their power source. While light and powerful compared to other types of batteries they are also renowned for exploding if not properly used. One sure way to get a Lipo battery to ignite is to attempt to draw current from it at a rate that exceeds its capacity. The maximum rate that a Lipo battery can supply power is called a C-Rating. A C-rating is calculated using the formula below:

$$\text{CRating} = \frac{\text{amps} \times 1000}{\text{mAh}}$$

Where amps is the maximum draw on the battery and mAh is the battery size.

Another sure way to set fire to a Lipo is to drain it completely. In actual fact the voltage levels in each Lipo battery cell should never be allowed to fall below 3.2 volts.

The flight time of a model aircraft (powered at full throttle for the entire flight) is calculated using the formula below:

$$\text{Time} = \frac{60}{\text{CRating}}$$

Where time is the flight time in minutes.

Given that the model will not be run at full throttle all the time, this time limit will include a reasonable safety factor. Lipo batteries are described by the number of cells, the mAh capacity and the C-Rating. Hence knowing the flight time and purchasing a battery with a high enough C-Rating will ensure no exploding model planes.

Design, in pseudo code, an algorithm which will input amps and mAh from the user and output the CRating and flight time to the user. Think carefully about the best way to employ sub modules in your algorithm. Your main algorithm should consist of only sub modules calls and, other than the main sub module, you should have at least three other sub modules. Note you may assume that the user input is always valid.

You can test your algorithm on the following inputs:

Amps	mAh
15	1800
8	800
27	2250
35	4000

Exercise Nine(Another Java implementation)
Convert your pseudo code design into a complete Java application.