The University of Melbourne
Department of Computer Science and Software Engineering
433-254 Software Design
Semester 2, 2003
Lab 1
Week 2

1. Introduction to the Sun Java Compiler

This lab serves as an introduction to the Sun Java compiler and virtual machine. This will be the *official* Java platform that we will be using in this course (for all the labs and the projects).

You should read through all the information below thoroughly, and follow any instructions given precisely. It is important that you understand everything so please ask your demonstrator for assistance if you have any problems.

- **Which machines do I use for Java?**

For all your Java programming, **you must only develop and run your code on the x86 machines (ie queeg, murree, mungee, murang)**

- **Setting up for compiling and running Java programs**

On the departmental student machines, various Java compilers and Java Virtual Machines (JVM) are installed. The official compiler and JVM we will be using (for project work particularly), is:

  - **Sun Java compiler**: **javac**
  - **Sun Java Virtual Machine**: **java**

These are two separate programs you run from the command line to compile and run your Java programs. You create and edit your Java programs using your favourite text editor (eg vim, emacs).

- **Setting up your environment**

Before you start using these, you must first ensure you have set your path correctly. To do this, type the following line (for bash):

export PATH=/usr/java1.4/bin:$PATH

This will set your path to the version we will be using.
Once you have done this, type the following:

java -version

It should report back the following:

java version "1.4.2"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2-b28)
Java HotSpot(TM) Client VM (build 1.4.2-b28, mixed mode)

If it does not, please ask your demonstrator to assist.

- **Using javac and java**

Now that you have your PATH set, you can test it.

(a) Get the file “HelloWorld.java” from the subject’s webpage.
(b) Once you have the file, type:

javac HelloWorld.java

If all is well, javac will compile the file with no messages. If errors exist, consult your lab demonstrator.

(d) Once compiled successfully, type ls to view the directory contents. You will notice the file HelloWorld.class which is the compiled Java Byte Code generated by javac.

(e) This is now ready for execution. You can run the program by typing:

java HelloWorld

(f) You should then see the message "Hello World!" appears on the next line.

(g) If this happened then congratulations, you have just compiled and executed your first Java program.

(h) Now modify the program in such a way that it displays your name, address and telephone number each on a separate line.

Aim: Understand writing to standard out.

2. Getting Additional Information

(a) Visit the 254 web page (http://www.cs.mu.oz.au/254).
(b) Read Information on the page.
(c) Go to the Useful links page and click on the Standard Class libraries link. This will be a useful reference when you start writing non trivial Java programs.
(d) Subscribe to and check the cs.254 newsgroup. This will be a major source for breaking news and important announcements so check it regularly (as with the web page).

3. Extra Work:

If you still have time and interested, try the following exercises that are more difficult!

(a) Write a java program that takes your first name and last name as command line arguments to the program and displays your name and last name on separate lines.

Aim: Understand the use of command line arguments.

(b) Write a program that calculates the total wages based on the number of hours worked. The wages are calculated at a rate of 8.25 per hour for hours less than 40 and at the rate of 1.5 the standard rate for any hours greater than 40. Number of hours is a command line argument to the program.

Hint: Use Integer.parseInt(String s) converts a string to an integer (To convert the number of hours from command line to integer). We will understand this as we get through the rest of the lectures and labs but use it for now.

Aim: Understand the use of if-else and constants.

(c) Write a program to take the student's grade as an input argument and print the comments as follows 100 (Perfect Score), 90-100 (Excellent), 80-90 (Good), 70-80 (Above Average), 60-70 (Average) 50-60 (Below Average), 0-50 (Not Passing). Hint: use switch statement.

Aim: Understand the use of switch statement.

(d) Write a program to print all odd numbers between 1 and 20. Note : Use while loop.

Aim: Understand the use of while loop.