CMPT 318 Assignment #1

Due: January 18, 1999, at the beginning of class.

Goals.
The objective of this assignment is to ensure that you have a working Java development environment, and to let you become familiar with working with Java applications and Applets.

Readings.

Read the following:

a. Chapters 1, 2, and 3, of the text (I will call this simply "Eckel").

b. Scan through your Java environment documentation until you are comfortable you know how to compile and run applications and applets. For the Sun JDK, the relevant documentation is Sun's "Tool Reference Pages".

c. The "Getting Started" Trail in Sun's "The Java Tutorial: A Practical Guide for Programmers". This book is online only and available at `http://java.sun.com/docs/books/tutorial/index.html`

   Work through the four main sections starting at `getStarted/index.html`

   (You may wish to download portions of this tutorial to have it online).

Questions.

(4 marks)

1. Write a program that prints the command arguments in reverse order. See Exercise #2 in Chapter 2 of Eckel. Hand in this program, along with your documentation.

(6 marks total)

2. Sun has 20 Java demo Applets included in its JDK distributions (they are available from the web at `http://java.sun.com/applets` if your JDK does not include them. Use the JDK 1.1 versions). It is perhaps easiest to run these applications by browsing them in your Java-enabled browser. Answer the following questions in short form---a few brief sentences is enough for each.

(2 marks)

a. Which of these applets could or could not have been *reasonably* implemented without programming, that is, with regular HTML interaction from a web server (e.g. movies, animated GIFs, sequence of pages, forms, etc.)? Briefly justify your answer.

(4 marks)

b. Glance at the source code for each of the 20 applets. You do not have to understand all of the code, but you should browse through it with interest. The sources for the applets are of varied lengths: some longer, but some shorter. But then again, some of the applets do more than the others.

   Name the three applets you think have the shortest code relative to the complexity of their behavior. That is, which three do you think do the most with the least amount of code? What is it about how they were programmed that make these applets so short? What makes the long ones longer?

(For reference, the applets are: Animator, ArcTest, BarChart, Blink, CardTest, Clock, DitherTest, DrawTest, Fractal, GraphicsTest, GraphLayout, ImageMap, JumpingBox, MoleculeViewer, NervousText, SimpleGraph, SortDemo, SpreadSheet, TicTacToe, WireFrame).

Note:
For all assignments you must document all of your code using javadoc. You must follow the assignment guidelines for the course (available on the CMPT 318 Homepage).