Coding Style Guidelines

We expect that you will write you programs so that they can be easily read by the instructors.

You should be using a strong visual layout that will make it easy for us to mark your code. You should be consistent with your use of indenting and you should space out your code both vertically and horizontally so that it is easy to read but don't get carried away.

We also expect you to use logical names for variables, constants, etc. It's okay to use x and y for loop variables, but for others refrain from using names like that when a longer name would be more easily understood.

Style Tips

The code you submit must contain the following in its body:

- **Routine Descriptions.**
  Each function/method must contain a couple of sentences describing the purpose of your routine. Your own routines should use descriptive names.
- **Data Structure Descriptions.**
  If you have defined any new data structure you must explain its purpose and comment on the nature of each field.
- **Block Comments.**
  You should be dividing your code into logical blocks within each routine. Each block must be accompanied by a comment that indicates what purpose the block serves.

Code Construction Resources

You can develop good coding style by reading what others have said about code construction ...

Although the following documents are written for C coding, the concepts are still applicable to Java.

An excellent guide for C-like coding style is the Indian Hill style guide. A number of versions of it exist -- the original, the U of T version, Keppel's version, and Brader's version. Brader's version appears to be the most complete and recent. The document is available in a number of formats.

Another useful guide is the GNU Coding Standards manual. This document covers a number of issues from interpreting command line options to documentation, and touches on coding style. This is a good document to peruse before starting to write your own Unix applications.

In terms of printed reference material ...

Microsoft Press has produced two books on the craft of programming: Steve Maguire's *Writing Solid Code*, and Steve McConnell's *Code Complete*. The former is a methodology of writing programs that discourage bugs; the latter, a collection of coding approaches, styles, and techniques for producing cleaner code and making better use of your time.

Code Complete is about 900 pages but well worth the read if you plan on programming for a career; *Writing Solid Code* is a much shorter book, written in a fun manner, chock full of useful programming tips and to make it truly irresistible, stories of how projects went wrong at Microsoft.