Java Programming Conventions

Terry Sergeant

Background

Every programming language has strict rules that must be followed. For some widely used languages there are programming conventions that arise. These conventions are not rules of the language itself but rather represent guidelines that many programmers voluntarily follow.

This document outlines some Java conventions that you should use in this course when writing Java programs.

Some Conventions

Variable Names

Variable names should begin with a lowercase letter and use "camel-case" (the first letter of each word is capitalized). Like this:

```
int     myFavoriteColor;
boolean isFinished;
String  bookTitle;
```

Method Names

Method names should begin with a lowercase letter and use camel-case. The method name should be a verb.

Classes

Class names should aways begin with an uppercase letter. Each class should appear in a separate source document having the same name as the class. For example, if you have written a class named `SpaceCadet` it should be in a file named `SpaceCadet.java` and it should be the only class in that file.

Indentation

Lots has been written about when to indent, etc. As long as you use a method I have seen before it will be fine. Each level of indentation should be at least 3 spaces. You should use only spaces or only tabs to indent (i.e., do not mix tabs and spaces).

Documentation

There is a handy tool called `javadoc` that will convert specially formed documentation into HTML format. For this course you should use the JavaDoc format as described below. NOTE: Adding a second asterisk when starting a comment is how you designate you are using JavaDoc notation.

```
/**
 * One sentence description followed by a period.
 *
 * @author  Your Name
 * @version Date of last change
 *
 * <p>
 * Longer description of program ...
 * </p>
 *
 * <p>
 * Which may require multiple paragraphs ...
 * </p>
 */
class Whatever
{
   /**
    * Description of classwide variable.
    */
   private int cool;

   /**
    * Each method gets a one sentence description followed by a period.
    *
    * @param first explain what the parameter first is for.
    * @param second explain what the parameter second is for.
    * @return explain what gets return by this method.
    *
    * <p>
    * If more needs to be said it can be done in paragraph form here ...
    * </p>
    */
   public boolean allDone(int first, double second)
   {
      int a;        // for local variables I prefer this form of documentation
      String ans;   // over JavaDoc conventions
      . . .
      . . .
      . . .
      return ans.equals("fun");
   }
}
```