WRITING JAVADOC COMMENTS

Javadoc is an industry-wide standard developed by Sun for commenting Java programs, and also a special tool included with the Java SDK which extracts these comments from your Java files to create documentation - these are the pages that you will see when you go to the Java APIs. The comments MUST be set off with the /** .... */ style of comments to be included by javadoc; other types of comments (both '/*....*/' and '//') are ignored. More information about Javadoc can be found in any Java textbook, and also at Oracle's Javadoc Home Page.

When you write your programs, you should put comments at the top with the filename of your program, your name, the date last modified, and a brief description (3-5 sentences) telling what the program does. Since the filename doesn't need to be added to the Javadoc description (it takes the name from the class declaration), use '//' comments to put the filename (and any other non-javadoc information) at the beginning of your page for easy identification. You can do this as follows:

```
// Student.java

/**
 * This class will create a student object, identified by name and
 * number, which may be used in simulations; the student will also
 * have an assigned status for class, and arrays to hold courses
 * taken and grades received, indexed in parallel.
 * (You may also include any assertions, assumptions, or other
 * comments here, including a note to your TF if you've still got any
 * kind of problem with the program.)
 *
 @author   Jan Jackson
 @version  Last modified 09_21_01
 **/
```

Your name should be designated with the @author tag; if you wish to include the date last modified, do so with the @version tag. Leave a blank line between the end of your description and the '@tags'. NOTE that these tags consist of the '@' symbol - no space after - followed by a specific label, then one or more spaces and the information related to the tag.

METHOD COMMENTS

Each method except *main* should have its own block of comments just **before** the declaration, set off by the `/** ......... */` style comment markers. You should include a 1-2 sentence summary of what the method does. For example, suppose I have a method to get two integers. This would be commented as follows:

```
/**
 * This method will use two ints declared in the
 * class, and will read in values from the user.
 */
public static void getNums() {
    Scanner in = new Scanner( System.in );
    System.out.print("Please enter the first number: ");
    num1 = in.nextInt();
    System.out.print("Please enter the second number: ");
    num2 = in.nextInt();
}
```

If your method passes any parameters, or returns a value, you will need to also include the javadoc tags for **parameters** or **return** values. NOTE that you must insert a blank line before the "@tags", and then list and describe them. **DO NOT** list local variables in these tags, only values which are included in the argument list or are returned. Return tags do not need the variable name. When we cover exceptions, they are also listed with an **@exception** tag IF THEY ARE THROWN AND NOT HANDLED (CAUGHT) IN THE CURRENT METHOD.

```
/**
 * This method will receive two ints passed in by the
 * user, and will return the larger of the two values.
 *
 * @param   val1    The first int value received
 * @param   val2    The second int value received
 * @return          The larger value of the two received
 */
public static int getMax(int val1, int val2) {
    if(val1 > val2){
        return val1;
    }
    else {
        return val2;
    }
}
```

List the parameter tags first, include the parameter's variable name, and a brief description of what the variable holds. The @return tag doesn't need a variable name, but should describe what is being returned. **You do NOT need @return tags for void methods, or @param tags for methods that don't receive any parameters.** Within the method, include short comments using the // marker, to indicate things that are being set or changed.

RUNNING JAVADOC

You may run Javadoc on your programs and look at the resulting documentation. It will produce several HTML files describing your classes based on the comments that you include.

- First, create a new folder at the same level as the folder containing your java files (or inside that directory) and name it `docs` (or whatever name you want).
- Second, in your teminal window, change into the directory containing your java files, and run the javadoc command to create the documentation pages:

  ```
  javadoc   -d   C:\pathToYour\docs  YourFileName.java
  ```

  If you used a different name than docs, substitute it above and include the correct path.
- Change into your `docs` directory, and click on the file you wish to see; it should open automatically in your browser, since it's an HTML file.