Computer Science 212
Object-Oriented Programming in Java

Lab 10

Aim: Program commentary and Javadoc.

As you should have learned, providing commentary in your source code is an essential part of the programming process. Each method (function, subprogram,…) and important section in your code (a while loop, a sequence of initializations) should have comments describing why that code is there (not *what* it does, but *why* it is there.)

Java takes this idea one step further, and has a convenient tool called *javadoc* for providing HTML (web page) code for other users of your Class's methods to use as documentation, similar to that found in the Java API library (hence, **Java doc**umentation).

Javadoc comments are included in your source code. So the compiler can tell the difference, *javadoc* comments start with a slash and double asterisks:

```
/*
    This is a normal comment in Java
*/
```

```
/**
    This comment is javadoc, and will be included in separate
    HTML (web page) format when the javadoc tool is run
*/
```

Javadoc notation includes several tags which start with the "at" sign ("@") so the compiler can recognize them and include them in the HTML document. Some common tags are:

`@params, @return`

Each parameter in a method should be documented to tell the reader what this parameter is for and what (if any) the return type is. As you've seen in the Java API, it is important to know what parameters a method needs. An example of using these tags is:

```
    /**
    * @param myFile  the file to be read
    * @param myChar  the character to count in the file
    * @return int    the number of characters in the file
    */
      public static int countChars(String myFile, char myChar) {
```

Working with Javadoc

Open Eclipse, create a Java Project for *Lab10* and import the file Lab10.java from the Z: drive into the *src* folder. Open the program.

Note that one of the methods, *fill Matrix* is preceded by javadoc statements. As part of the process of exporting javadoc, Eclipse needs to be configure (once) to know where the javadoc tool is:

- Right click on your Java project *Lab10*
- Choose **Export | Java | Javadoc | Next**
- At the top where it says "Javadoc Command," click **Configure**
- Now you need to browse to the folder that contains the javadoc tool. In the lab this should be `c:\Program Files\Java\jdk1.6.0_12\javadoc.exe`
- Choose OK when you have highlighted the file javadoc.exe
- The rest of the default settings are good, click **Finish**

You should see output in the Console window showing the javadoc tool is generating the HTML code for the documentation. Once its done, you can view the javadoc:

- Open a web browser.
- Choose File | Open, and browse the file system starting at H:\ and going to your Eclipse workspace then to the folder for Lab 10, then to the directory **doc** (a directory created by Eclipse for the javadoc, and open **index.html** (the root file for the HTML documentation.)

Now go back to your Lab10 program and continue entering javadoc for the other methods. Eclipse makes this easier because it will generate parts of the javadoc it can figure out:

Place the cursor on the line above the method heading

```
   public static int sumMatrix(int[][] m){
```

Type "/**" and <enter>, and you will see Eclipse fill some javadoc tags. Right after each tag, type a description of what that parameter means, such as "the matrix to be filled":

```
    * @param m The matrix to be filled
```

Fill in the rest of the tags.

Now you can export the javadoc again (Eclipse will warn you that you are overwriting the *doc* directory – that's OK.)