Javadoc Notes (Version 1.1)
BE-104, Spring '05, Dr. C. S. Tritt

Javadoc is an automatic documentation generation system often used for Java programming projects (and packages). In fact, the "official" Java documentation is generated using javadoc. Javadoc generates a collection of .html (web page) documents based on information it extracts from java code source code files. Some of this extracted information is based the contents of javadoc style comments (including the values of included javadoc "tags").

Typical javadoc comments locations and contents include:

> Before the *class* statement: A general description of the class. This comment should include @author and @version tags.
>
> Before each member variables: A description of the variable. Do not include the variable type (it will be shown automatically). Do provide units if applicable.
>
> Before each method definition: A description of what the method does (but generally not how it does it). These descriptions generally start with verbs. The comments don't have to specify that particular methods are constructors since javadoc will figure this out and treat them accordingly. These comments should include @param, @return and @throws tags as appropriate.

In general, keep javadoc comments short and succinct.

Extra spaces, line breaks and leading stars in javadoc comments are ignored. Javadoc comments can include embedded html tags like <br>, <p>/</p>, <b>/</b>, <i>/</i>, etc.

Selected Tags and Their Use

@author *name* – Specifies the author of the class.
@version *version* – Specifies
@param *parameter description* – Describes each parameter (include units but not types).
@return *description* – Describes what is returned by the method (include units but not types).
@throws *exception* – Specifies the exception(s) thrown by the method.

Tags apply to all text to the next tag or the end of the javadoc comment so they generally should be placed at after other comment text.

Javadoc documentation can be generated from within the Eclipse environment. To do this, use *File* | *Export…* | *Javadoc* to initiate a Wizard like sequence of dialogs. These dialogs allow specification of the javadoc command, destination folder for the documentation and which visibilities of class members to document (the default is public only) among other things.