

Lab 9 – Javadoc

Mr. P. Guglielmi
William Lyon Mackenzie CI
ICS4U

April 14, 2024

Contents

1	Java Documentation	1
1.1	Tags	1
1.1.1	@param Format	1
1.1.2	@return Format	1
1.2	Comment Format	1
1.3	Generating the Documentation.	2

1 Java Documentation

Javadoc is an automated tool for creating documentation for a class or program. The tool will generate an HTML document which will outline and describe the fields and methods contained within a class. One document will be created for each class in your program. Comments must be written in a specific format and use specific tags for the tool to pick up and use to create the documentation.

1.1 Tags

Tags specify different information, such as the author, method parameters, or return values. Some key tags that need to be used are:

- `@Author` – Specifies the author. Required, in top comments
- `@Version` – Specifies the version. Required, in top comments
- `@param` – Used to specify the parameters for constructors and methods. One tag required for each parameter.
- `@return` – Used to specify the return value. Required in for non-void methods

1.1.1 `@param` Format

The param tag needs the name of the parameter specified, followed by a space and the description. The description should start with a lower case letter, and describe the data type.

Example:

```
@param x an integer which will be computed
```

1.1.2 `@return` Format

Return functions similarly to `@param`, but does not require that a variable be specified, just the description.

```
@return returns the computed value
```

1.2 Comment Format

A javadoc comment opens with `/**` and ends with `*/`. All items between will be read and included in the documentation. The general format it:

1. Description
2. `@param` tags (if method has parameters)
3. `@return` tag (if method returns a value)

```
/**
This is the method description
@param x the int input for the method
@return an integer which is the computed value
*/

public int myMethod (int x){
//method implementation
return value;
}
```

1.3 Generating the Documentation.

Most IDEs have a tool for running javadoc and generating the documentation. In jGrasp, this is found in the project menu. The classes must also be in a project structure in order to generate the javadoc. Other IDEs will have similar functionality. The finished javadoc will be called *Classname.html*, and Javadoc will generate many additional files which are generally not needed for smaller projects.

2 Activity and Deliverable

Take one of the example programs on the class GitHub (say, Sorting or Searching) and write the javadoc comments and generate the documentation. Submit the resulting html files to Brightspace.

Marks: 10C