Javadoc Overview

- Javadoc is a tool for creating html documentation
- The documentation is generated from comments
- It produces actual html web pages
- Helps keep documentation consistent with the code
Doc Comments

- Doc comments start with /** and end with */.
- The /** and */ should be on their own lines.
- Additional lines need to start with a *.
- Comments can have html tags.

```java
/**
 * This is a javadoc comment
 */
```
Javadoc Tags

- Use tags to help Javadoc parse your comments.
- Tags start with a `@` and are case sensitive.
- It must be at the beginning of the line

```java
/**
 * Prints the kth element of the list
 *
 * @author Alexander Lee
 * @param list The list who’s element is to be printed
 * @param k The index of the item to be printed
 */
public void printK(List list, int k) {
}
```
@link package.class#member label

This tag inserts a link that points to the documentation of the specified class.

label represents the text that shows up.

The curly brackets indicate that it is an inline tag.

So it can be placed anywhere in the comments, doesn’t have to be in the beginning.

```java
/**
 * Get the names of an object
 *
 * @deprecated This function should not be used
 * Use {@link #getFirstName()} and
 * {@link al91.Person#getLastName()} instead.
 */
```
Important Tags

Some important tags

- @author: describes the author
- @param: describes a specific parameter
- @return: describes the return value
- @throws: describes exceptions it throws
- @deprecated: describes the reason
- @see package.class#member
Javadoc and Eclipse

- To generate the doc, go to Project and Generate Javadoc

- Eclipse automatically generates Javadoc comments if the method signature is already written.

- Can configure Eclipse to complain about missing Javadoc by going to preferences, compiler, javadoc
Design a GPS system that finds the shortest route from the current location to the destination.

- There are three ways of transportation, by car, by public transportation and by walking.
- The GPS system should be able to return a route given any current location, and destination
- Can be easily updated to add new roads, or to avoid roads
Debug Exercise

Run Lab02 in the debug perspective. The program currently prints out a grid of 4’s, but this is not right. Try to debug it using:

- breakpoints
- stepping over, in, out
- continuing
- inspecting variables