Code Comments and Documentation

How to create an API for your code.
Life Cycle Documentation

- Take Software Engineering to see how businesses document the product life cycle.
  - Business requirements.
  - Functional specification.
  - Design document.
  - Implementation document.
Your code can be automatically converted into API pages.

Javadoc.

And similar utilities for some other languages.

Converts your code comments into API pages!
Using //... Versus /** ... */

- Use /** ... */ to write a description of a class, any constructor, any method, or public instance variable.
  - put only directly above the class, constructor, method, or instance variable.
  - These comments will show up in the API!

- Use //... for everything else
  - E.g., use within methods or other code.
  - These comments will NOT show up in the API!
How To Write Comments

Write comments this way.

1. Make the first sentence a non-trivial summary of the class or method.
2. Put the comments directly before the class, method, constructor, or instance variable.
3. Wrap comments in

    /**
     * Simulator for an artificial ecosystem. Coolest class ever written.
     * Blah, blah, blah.
     */

    public class Ecosystem
    {
    ...}
Feel free to include simple html.

- `<p>` to separate paragraphs.
- `<code>` …some code… `</code>`

```java
/**
 * Simulator for an artificial ecosystem. Coolest class ever written.
 * Blah, blah, blah.
 * <p>
 * More blah, blah… `<code>` Ecosystem eco = new Ecosystem() `</code>`
 */
public class Ecosystem
{
...
```

- Some html may mess up the javadoc formatting. Be sure to check.
Comments Can Include Tags

Tags add special sections to the API.

- **@author** Your Name
  - adds an author paragraph
- **@param** parameterName description
  - adds description of input parameter
- **@return** description
  - adds description of what is returned (the type is added automatically).
- **@throws** exceptionClassName description
  - adds what exception may be thrown and description of why.
- **@version** description
  - adds version of the code. e.g., “Coolest Code Ever 4.5”
- **@see** link
  - adds “See Also:” section
- others
Style of Comments

- For accepted style conventions, see
  - http://java.sun.com/j2se/javadoc/writingdoccomments/

- If follow all the conventions, your API will look very professional.

- In general, don’t say the obvious.
  - @param channel The channel.
    - BAD
  - @param channel Current channel selected on the TV.
    - GOOD
Generating the API

- Just type the following at the command prompt
  - from same folder that contains your classes
    
    javadoc className.java

- Can add options that include the author and version
  - not included by default
  - javadoc className.java -author
  - javadoc className.java -version
  - javadoc className.java -author -version -public
    - shows only public classes (with the version and author)
Generating API For Entire Package

- `javadoc packageName1`

- `javadoc packageName1 packageName2`

- `javadoc -d c:/html packageName`
  - saves all the API docs in the specified directory (-d) (in this case c:/html)

- `javadoc C:/myCode/packageName`
  - Don’t have to be in the same directory to use javadoc.
  - Just specify path to file.
For More Info

- [http://java.sun.com/j2se/1.4.2/docs/tooldocs/solaris/javadoc.html](http://java.sun.com/j2se/1.4.2/docs/tooldocs/solaris/javadoc.html)

- [http://java.sun.com/j2se/javadoc/writingdoc comments/](http://java.sun.com/j2se/javadoc/writingdoc comments/)