Class Documentation

Chris La Pilla
CompSci 6
Summer 2007

Documentation

• In order to make code more useful, we need to provide documentation, i.e. text in addition to the code that explains what it does
• Documentation acts as a reference for other programmers who want to use your code, but also for you
• Documentation can serve two purposes:
  – Explain how the code works (implementation details)
  – Explain how to use it (interface)

Comments (// or /* */)

• These comments allow you to place documentation directly in the method bodies
• Their goal is to describe how pieces of the program work
• All of your programs should be commented
• It’s a good idea to place a comment before a group of lines that are all related to the same task or computation
• Don’t overdo it:
  – No need to comment each line unless the line is particularly hard to understand
  – Don’t comment lines that are self-explanatory

Example

What does this code do?

```java
while(num2 != 0)
    temp = num2;
    num2 = num1 % num2;
    num1 = temp;

System.out.println(num1);
```

Example

//Euclid's algorithm for computing the Greatest Common Factor (GCF) of num1 and num2
while(num2 != 0)
    temp = num2;
    num2 = num1 % num2;
    num1 = temp;

System.out.println(num1); //print the GCF

• Is this helpful?
Javadoc Utility

- In addition to comments that go inside method bodies, there are special documentation comments that are used to describe entire classes, methods, or variables
- These comments begin with /** and have a special format
- Primary purpose of these types of comments is to describe how to use the code without having to actually look at the source

Javadoc

- When run, the Javadoc utility will extract the information in these comments and create web pages
- You can then refer to the web pages without having to look at the source code
- Eclipse also has built-in support for Javadocs
- See exampleJul09_StudentGrades on snarfer site for example