CMSC 131

Object-Oriented Programming I

Libraries, Round Off Errors

Dept of Computer Science
University of Maryland College Park

This material is based on material provided by Ben Bederson, Bonnie Dorr, Fawzi Emad, David Mount, Jan Plane

Overview

- Floating Point Calculations
- Libraries
- Java API

Project Documentation

- A general comment at the top of every class
- A detailed comment above each method (describing the "contract" for
  the method... pre-conditions, post-conditions, etc.)
- For MANY variables, a comment should appear near the declaration
- Frequently comments should be embedded in the midst of complex code fragments to guide the reader about what is going on
- As already mentioned avoid code duplication

Floating Point Calculations

What will this print?

```
public class SimpleMath {
    public static void main(String[] args) {
        if (3.9 - 3.8 == 0.1) {
            System.out.println("I am a very smart computer.");
        } else {
            System.out.println("I can't do simple arithmetic.");
        }
    }
}
```

- What is the output?
- Floating point numbers in Java are stored in binary representation, and frequently numbers that are easily represented in base 10 cannot be represented precisely in base 2
- What can we do?

Floating Point Calculations

Two important rules:

- You can never use == to compare floating point values. Instead, check if two numbers are within a certain tolerance of each other.
  - `Math.abs((3.9 - 3.8) - 0.1) < EPSILON`
- Never use floating point values to represent money, e.g., 3.52 to represent $3.52. Instead, use integer 352 to represent 352 pennies.

Libraries in Java

- Library → implementation of useful routines that are shared by different programs
- Java mechanism for creating libraries: packages
  - Package: group of related clases
  - Example: java.util (contains Scanner class)
- To create a package in Eclipse use
  - File→New→Package
- To use a class from a package, you can use a fully qualified name (package name + class name):

  java.util.Scanner s = new java.util.Scanner(System.in);
- You can also import the class in the beginning of the file

  import java.util.Scanner;
- To import class in a package:

  import java.util.*;

  (Imports Scanner as well as other classes in package)

Package java.lang

- A special package containing widely used classes:
  - `String`
  - `Math`
  - etc.
- `java.lang.*` is *automatically imported* by every Java program

Package Management

- A class can be added to a package by including:

  package <name of package>;

  in source file (usually very first line)
- The variables / methods provided by a class / package are often called its API (= Application Programmers Interface)
- APIs should be documented
- java.lang documentation:
  http://download.oracle.com/javase/6/docs/api/java/lang/package-summary.html
- We can use the javadoc utility to generate API documentation. Let's generate example for some code we have written

String API & Math API

- Java API
  - http://download.oracle.com/javase/6/docs/api/
  - You should have this in your bookmarks
  - You can even download it to your computer
  - Extremely helpful
- String API
  - http://download.oracle.com/javase/6/docs/api/index.html
  - Implements lots of string functions
- Math API
  - http://download.oracle.com/javase/6/docs/api/index.html
  - Implements lots of Math functions