



CMSC 131

Object-Oriented Programming I

Switch, Rounding Errors, Libraries

**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

Accessing Private Instance Variables

- **IMPORTANT:** Remember that methods of a class can access private elements of parameters that belong to the same class
- Assume we have a `Student` class with a private instance variable called **`name`** and a method called `checking`

```
public boolean checking(Student p) {  
    // We can access p.name here even though is private  
    // We don't need a get method. The method checking  
    // can access its own private instance variables and  
    // also those of objects that belong to the same class  
}
```

Ternary Operator

- We can rewrite your maximum method by using the ternary operator:

Expr ? exprValueIf**Expr**IsTrue : exprValueIf**Expr**IsFalse;

- Rewriting maximum

int maximum = x > y ? x : y;

Switch Statement

- You can use a switch statement instead of a cascaded if statement if the expression you are testing is an integer, string or enum type (to be seen later on)
- **Example:** SwitchExample.java

About Naming Constants

- If a constant is static then use uppercase letters

*final **static** int **MAX** = 10;*

- If a constant is not static then **do not** use uppercase letters
 - Use camel case

*final int **maxPressure** = 50;*

- **Example:** ScienceExperiment.java

Floating Point Calculations

- What is the output of the following code:

double difference = 3.9 - 3.8;

- **Example:** FloatCalculations.java
- 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 shared by different programs
- Java mechanism for creating libraries: **packages**
 - Package: group of related classes
 - Example: java.util (contains Scanner class)
- To create a package in Eclipse use

File→New→Package

Libraries in Java

- To use a class from a package you can use a **fully qualified name**
- 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 in the source file (usually very first line):

package <name of package>;

- The variables/methods provided by a class/package are often called its **API** (= Application Programmers Interface)
- APIs should be documented
- **java.lang** documentation:
<http://docs.oracle.com/javase/8/docs/api/java/lang/package-summary.html>