



CMSC 131

Object-Oriented Programming I

Access Specifiers, Get/Set Methods, **Midterm Review**

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

- ▶ Access Specifiers
- ▶ Get/Set Methods
- ▶ General Class Organization
- ▶ Midterm Review

Video

- ▶ History of Computing in 147 seconds
 - <http://www.computerweekly.com/Articles/2009/08/17/237346/Video-A-history-of-computing-in-147-seconds.htm>

Access Specifiers

- ▶ We have seen public
- ▶ Two points of view that help us understand these specifiers
 - Programmer
 - User
- ▶ Let's see package
 - All classes in the same package can access members defined as package
- ▶ Let's see private
 - Only methods of the class can access private elements (instance variables/methods)
 - Private methods are methods that provide support to other methods and not intended for users of the class
- ▶ Let's see an example based on get/set methods
 - If we define a field as private then we may need to provide methods that allow us to access (get method) or modify (set method) the field.
 - A set method can control the validity of data passed in

About main

- ▶ Every class can have a static main method
- ▶ It can be used for testing methods of the class
- ▶ Keep in mind that this method may not behave the same if place in another class
 - Let's see an example of a main method for a class that is copied to another class (driver)
 - Our main method accesses private elements
- ▶ Careful about main definition
 - It needs to have void as return type
 - It needs to be static otherwise you will not be able to run the application

Static vs. Non-Static Methods

- ▶ When should a method be defined static?
 - When it makes no references to the class instance variables
- ▶ Why would I like to define a method as static?
 - Because you don't need to create an instance of the class in order to call it
 - Notice you can still call the method through a instance of the class
- ▶ Keep in mind that
 - Static methods can only call static methods
 - Static methods can not reference instance variables
- ▶ Let's see some examples where static methods try to access instance variables and other methods

General Class Organization

- ▶ Constructors
 - Default constructor needed?
- ▶ Get/Set Methods
 - Notice that not every class needs to define get/set methods
- ▶ Class specific methods
- ▶ equals method
- ▶ static methods
- ▶ main method

Midterm Review

- ▶ Variables
- ▶ Conditionals
- ▶ Loops
- ▶ Static methods
- ▶ Others