

# Lecture 12:

# Static Methods and Variables

Last time:

1. Project #2 Hints
2. Aliasing
3. Constructors, Accessors, Mutators
4. Equality
5. Printing an object
6. `for` loops

Today:

1. `for` loops (from last lecture notes)
2. Static variables and methods



# Static Variables and Methods

- We have seen how to declare:
  - Instance variables in classes:

```
public int tokenLevel = 3;
```
  - Methods in classes:

```
public void setName (String nameDesired) {  
... }
```
- Objects created from a class receive their own copies of instance variables and methods
- Java also has **static variables and methods**, which are shared by all objects in a class

# Why Have Static Variables / Methods?



- Sometimes info needs to be shared among all objects in a class
  - How many objects in a class have been created?
  - A constant that needs to be the same
- Sometimes it is useful to have methods that can be invoked without first creating objects
- We will see how static components help

# Declaring Static Methods (and variables and constants)



- Static methods

```
public static void main (...) { ... }  
public static void drawLetter(DrawingGrid  
    grid, char letter, Color color) { ... }
```

- How do we call static methods?

```
LetterMaker.drawLetter(grid, choice.charAt(0), color)
```

- Can have static variables and constants too

```
public static int numStudents = 0;  
public static final int MAX_ENROLLMENT = 0;
```

- How do we use static variables and constants? (see next example)

```
StudentRoster.numStudents  
StudentRoster.MAX_ENROLLMENT
```

# Example: Object Counting

```
public class StudentRoster {
    public static int numStudents = 0;
    public static final int MAX_ENROLLMENT = 50;
    public static int getNumStudents () {
        return numStudents;
    }
}
...
```

```
public class Student {
    <code from previous Student class>
    Student (String newName) {
        name = newName;
        numStudents++;
    }
}
```



# What Is Printed?

- ```
Student s1 = new Student ("John Doe");  
Student s2 = new Student ("Mary Roe");  
System.out.println (StudentRoster.getNumStudents());
```

2
- ```
Student s3 = new Student ("Eduardo Duhalde");  
System.out.println (StudentRoster.getNumStudents());
```

3
- ```
System.out.println (StudentRoster.MAX_ENROLLMENT);
```

50

# Class Access to Static Variables and Methods



- If  $C$  is a class,  $sv$  is a static variable, and  $sm$  is a static method ...
- Then  $sv$ ,  $sm$  can be accessed via:
  - $C.sv$
  - $C.sm$
- I.e. no object in  $C$  needs to be created!

# When To Use Static Variables?

- Class-wide constants

```
static final int MAX_ENROLLMENT = 50;
```

- Class-wide aggregate data

```
static int numStudents = 0;
```

# When To Use Static Methods?

- When a method should be invocable without object creation
- When a method should not change instance variables
  - A static method can only change static variables
  - Instance variables can only be changed by non-static methods