

CSSE 220 Day 2

Class, Objects, and Methods in Java
UML Class Diagram Basics

Your questions about ...

- ▶ The syllabus
 - ▶ Java
 - ▶ etc.
-
- ▶ Could everyone checkout and commit the HW1 project?

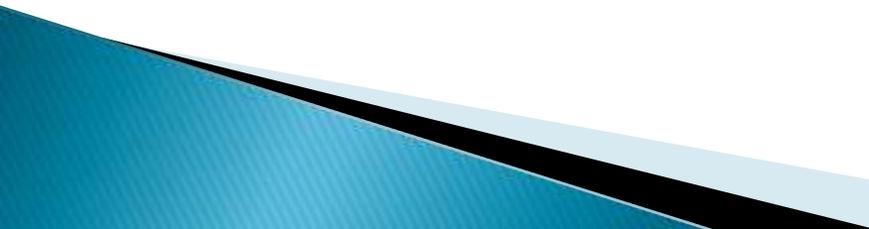
How's the Wiki Working?

- ▶ Is editing going OK?
- ▶ What about readings that don't cover the whole chapter?

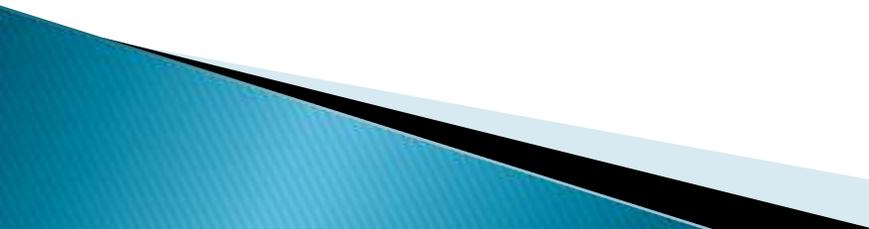
Bonus points for reporting bugs

- ▶ In the textbook
 - ▶ In any of my materials.
 - ▶ Use the Bug Report Forum on ANGEL
 - ▶ More details in the Syllabus
-
- ▶ Subscribe to the discussion forums on ANGEL
- 

Some major emphases of 220

- ▶ Reinforce and extend OO ideas from 120
 - Major emphasis on inheritance
 - GUI programming using Java Swing
 - ▶ Object-oriented Design
 - ▶ Data Structures
 - Introduce Algorithm efficiency analysis
 - Abstract Data Types
 - Specifying and using standard Data Structures
 - Implementing simple data structures (lists)
 - ▶ Recursion
 - ▶ Simple Sorting and searching
 - ▶ A few additional Software Engineering concepts
- 

What will I spend my time doing?

- ▶ Small programming assignments in class
 - ▶ Larger programming problems, mostly outside of class
 - Exploring the JDK documentation to find the classes and methods that you need
 - Debugging!
 - Reviewing other students' code
 - ▶ Reading (a lot to read at the beginning; less later)
 - Thinking about exercises in the textbooks
 - Some written exercises, mostly from the textbook
 - ▶ Discussing the material with other students
- 

Identifiers (Names) in Java

- ▶ The rules:
 - Start with letter or underscore (_)
 - Followed by letters, numbers, or underscores
- ▶ The conventions:
 - variableNamesLikeThis
 - methodNamesLikeThis()
 - ClassNamesLikeThis

Variables in Java

- ▶ Like C:

- `int xCoordinate = 10;`

- ▶ But Java catches some mistakes:

- `int yPosition;`

- `printf("%d", yPosition);`



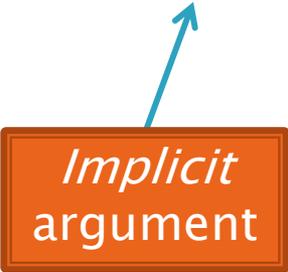
What does this do in C?

- ▶ Java will detect that **yPosition** isn't initialized!

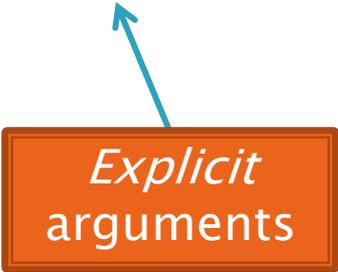
Using Objects and Methods

- ▶ Works just like Python:
 - *object.method(argument, ...)*

Implicit
argument



Explicit
arguments



- ▶ Java Example:

```
String name = "Bob Forapples";  
PrintStream printer = System.out;
```

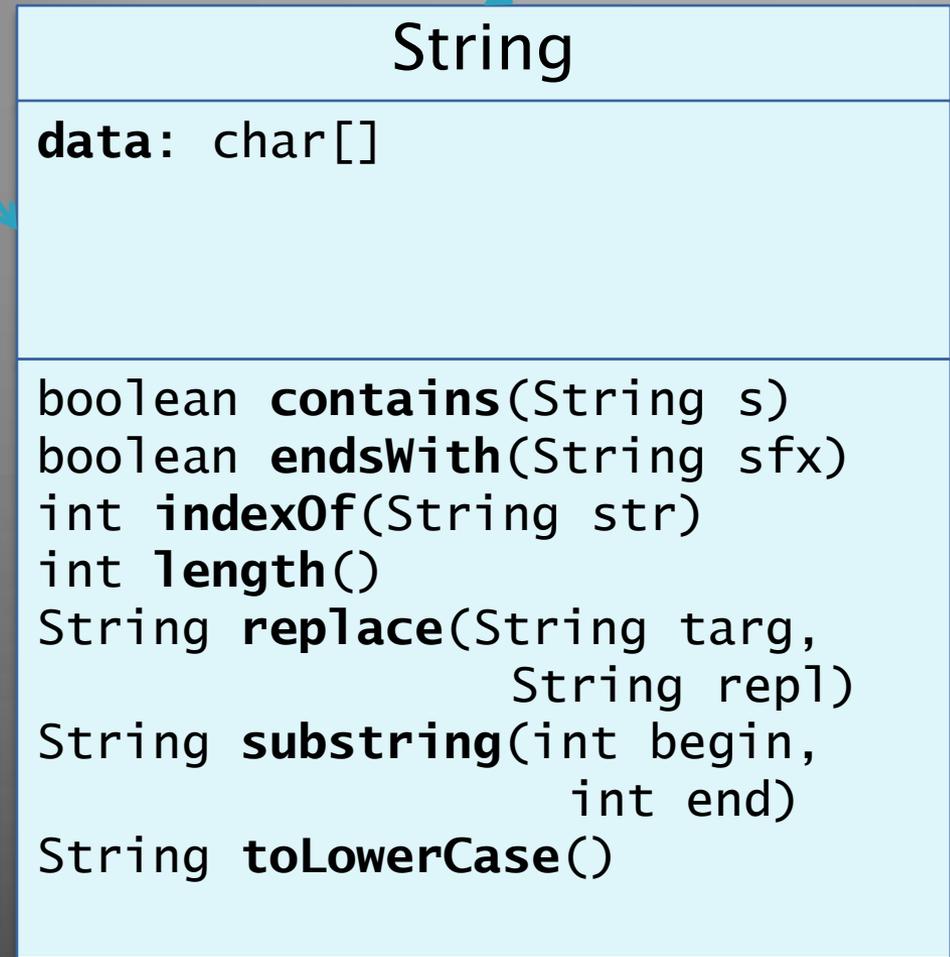
```
int nameLen = name.length();  
printer.printf("'%s' has %d characters", name, nameLen);
```

Separating Use from Implementation

- ▶ Can use methods of an object without knowing how its implemented
 - Recall zellegraphics from 120: `Line.setWidth(5)`

UML Class Diagram

- ▶ Shows the data (fields) and methods of the objects of a class
- ▶ Does *not* show the implementation
- ▶ Not necessarily complete



Class name

Fields

Methods

Exercise

- »» Checkout ObjectsAndMethods from SVN
Work on UsingStrings.java

Interlude – From worldmag.com

WORLD
MAGAZINE

Archives *1996 to the Present*

Quick Takes

ODDBALL OCCURRENCES

Bovine force

» If Linda and Charles Everson Jr. had been driving just a bit faster, they may not have celebrated another anniversary. While celebrating their first anniversary, the Michigan couple was driving on Highway 150 alongside a cliff near Manson, Wash., when something fell from above and crushed the hood of their minivan. Instead of falling rocks, it was a falling cow. The 600-pound cow, which had fallen from 200 feet up, crushed the front of their minivan, but the couple escaped unscathed. The 1-year-old bovine wasn't so fortunate.

wenatchee**w**orld.com
THE FIERCELY INDEPENDENT VOICE OF NORTH CENTRAL WASHINGTON

The cow heard around the world Media milking falling-heifer story to death

The couple were driving back to their Manson hotel on Highway 150 Sunday after attending a church service in Chelan when the 600-pound heifer named Michelle dropped from above and landed on the hood of their Buick Terraza and bounced off onto the road. Everson said he was stunned and kept on driving, repeating to himself "I don't believe it. I don't believe it."



Passing Parameters

- ▶ Arguments can be any expression of the “right” type
 - See example...
- ▶ What happens if we try to give `substring()` an explicit argument that isn't a number?
 - How does compiler know that `rhit.length()` evaluates to a number?
 - What's the return type of `length()`?
- ▶ Static types help compiler catch bugs.
 - Important in large programs

Primitive types

Primitive Type	What It Stores	Range
byte	8-bit integer	-128 to 127
short	16-bit integer	-32,768 to 32,767
int	32-bit integer	-2,147,483,648 to 2,147,483,647
long	64-bit integer	-2^{63} to $2^{63} - 1$
float	32-bit floating-point	6 significant digits (10^{-46} , 10^{38})
double	64-bit floating-point	15 significant digits (10^{-324} , 10^{308})
char	Unicode character	
boolean	Boolean variable	false and true

figure 1.2

The eight primitive types in Java

Most common
number types in
Java code

Exercise

»» Work on SomeTypes.java

Constructing Objects

left, top, width, height



- ▶ Example:

```
Rectangle box = new Rectangle(5, 10, 20, 30)
```

- ▶ Several steps are happening here:

1. Java reserves space for a `Rectangle` object
2. `Rectangle`'s *constructor* runs, filling in slots in object
3. Java reserves a variable named `box`
4. `box` is set to refer to the object

Accessors and Mutators

▶ Accessor methods

- Get a value from an object
- Examples:
 - `box.getHeight()`
 - `box.getWidth()`

▶ Mutator methods

- Change the *state* of an object (i.e., the value of one or more fields)
- Examples:
 - `box.translate(10,20)`
 - `box.setSize(5,5)`

Exercise

- »» Finish quiz
- Continue working on homework