

# CSSE 220 Day 9

Arrays, ArrayLists,  
Wrapper Classes, Auto-boxing

Check out *ArraysAndLists* from SVN

Questions?

# Exam Coming!

- ▶ Test next Wednesday
- ▶ Topics from Ch. 1–7
- ▶ Will include:
  - A paper part—logic, short answer, fill-in-the-blank
  - A programming part—a few small programs, unit tests provided
- ▶ Review in class Monday
  - Bring questions
  - I won't prepare anything but am happy to cover whatever you want, including working examples

# Array Types

- ▶ Syntax: *ElementType*[] *name*
- ▶ Examples:
  - A variable: `double[] averages;`
  - Parameters: `public int max(int[] values) {...}`
  - A field: `private Investment[] mutualFunds;`

# Allocating Arrays

- ▶ Syntax: **new *ElementType*[length]**
- ▶ Creates space to hold values
- ▶ Sets values to defaults
  - **0** for number types
  - **false** for boolean type
  - **null** for object types
- ▶ Examples:
  - **double[] polls = new double[50];**
  - **int[] elecVotes = new int[50];**



Don't forget  
this step!

# Reading and Writing Array Elements

- ▶ Reading:

- `double exp = polls[42] * elecVotes[42];`

Sets the value  
in slot 37.

Reads the element  
with index 42.

- ▶ Writing:

- `elecVotes[37] = 11;`

- ▶ Index numbers run from 0 to array length - 1

- ▶ Getting array length: `elecVotes.length`

No parens, array length  
is (like) a field

# Arrays: Comparison Shopping

Arrays...	Java	C	Python
have fixed length	yes	yes	no
are initialized to default values	yes	no	n/a
track their own length	yes	no	yes
trying to access “out of bounds” stops program before worse things happen	yes	no	yes

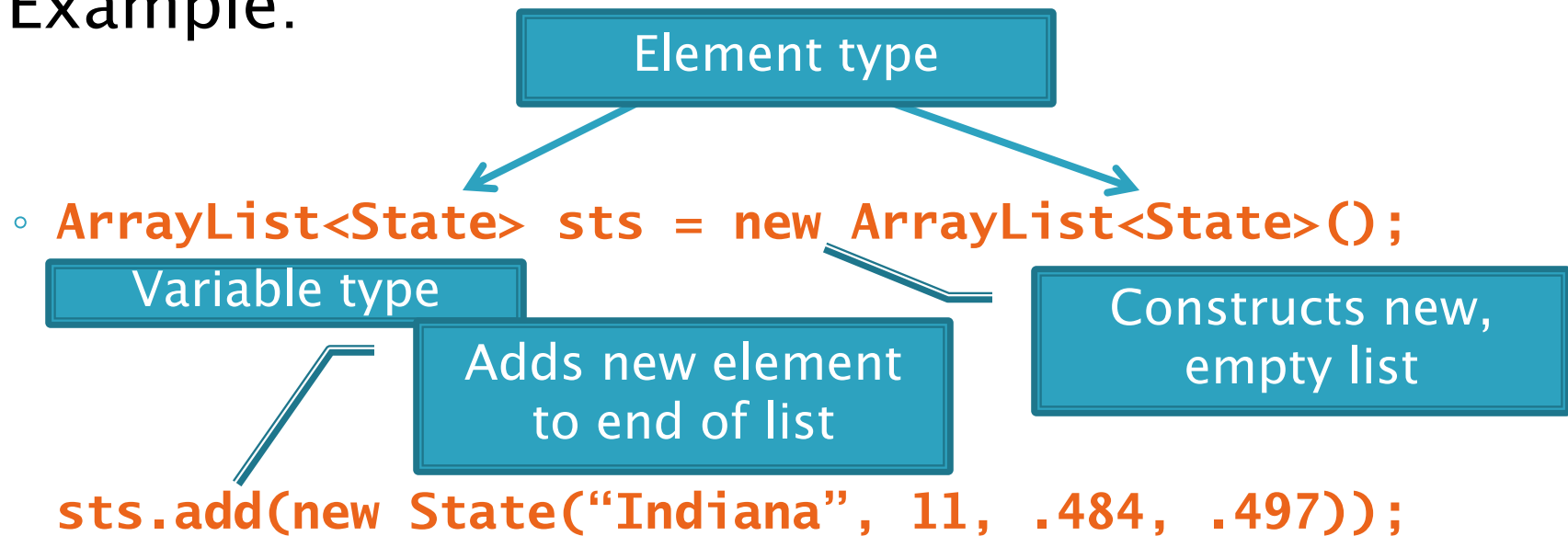
# Live Coding

- »» Begin ElectionSimulator program



# What if we don't know how many elements there will be?

- ▶ ArrayLists to the rescue
- ▶ Example:



- ▶ **ArrayList** is a *generic class*
  - Type in <brackets> is called a *type parameter*

# Array List Gotchas

- ▶ Type parameter can't be a primitive type
  - Not: **ArrayList<int> runs;**
  - But: **ArrayList<Integer> runs;**
- ▶ Use get method to read elements
  - Not: **runs[12]**
  - But: **runs.get(12)**
- ▶ Use size() not length
  - Not: **runs.length**
  - But: **runs.size()**

# Lots of Ways to Add to List

- ▶ Add to end:

- `victories.add(new WorldSeries(2008));`

- ▶ Overwrite existing element:

- `victories.set(0, new WorldSeries(1907));`

- ▶ Insert in the middle:

- `victories.add(1, new WorldSeries(1908));`

- Pushes elements at indexes 2 and higher up one

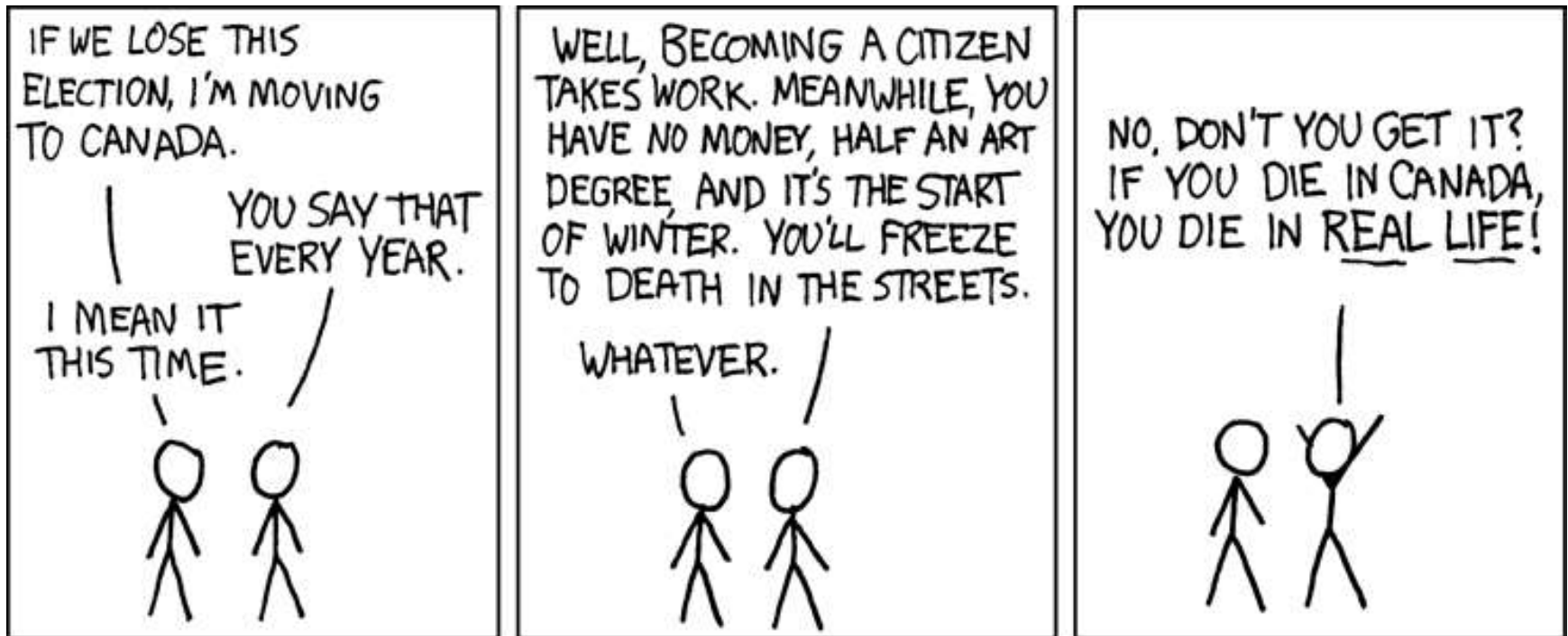
- ▶ Can also remove:

- `victories.remove(victories.size() - 1)`

# Live Coding

- »» Convert ElectionSimulator to use ArrayLists

# Cartoon of the Day



IT'S ALL REAL!

# So, what's the deal with primitive types?

## ▶ Problem:

- ArrayLists only hold objects
- Primitive types aren't objects

## ▶ Solution:

- *Wrapper classes*—instances are used to “turn” primitive types into objects
- Primitive value is stored in a field inside the object

Primitive	Wrapper
byte	Byte
boolean	Boolean
char	Character
double	Double
float	Float
int	Integer
long	Long
short	Short

# Auto-boxing Makes Wrappers Easy

- ▶ Auto-boxing: automatically enclosing a primitive type in a wrapper object when needed
- ▶ Example:
  - You write: **Integer m = 6;**
  - Java does: **Integer m = new Integer(6);**
  - You write: **Integer ans = m \* 7;**
  - Java does: **int temp = m.intValue() \* 7;**  
**Integer ans = new Integer(temp);**

# Auto-boxing Lets Us Use ArrayLists with Primitive Types

- ▶ Just have to remember to use wrapper class for list element type
- ▶ Example:
  - `ArrayList<Integer> runs = new ArrayList<Integer>();`  
`runs.add(9); // 9 is auto-boxed`
  - `int r = runs.get(0); // result is unboxed`



# Enhanced For Loop and Arrays

- ▶ Old school

```
double scores[] = ...
double sum = 0.0;
for (int i=0; i < scores.length; i++) {
    sum += scores[i];
}
```

- ▶ New, whiz-bang, enhanced for loop

```
double scores[] = ...
double sum = 0.0;
for (double sc : scores) {
    sum += sc;
}
```

Say "in"

- No index variable
- Gives a name (sc here) to each element

# Enhanced For and ArrayLists

```
▶ ArrayList<State> states = ...  
  int total = 0;  
  for (State st : states) {  
      total += st.getElectoralVotes();  
  }
```

# Live Coding

»» Finish ElectionSimulator