CSE115 / CSE503
Introduction to Computer Science I

Dr. Carl Alphonce
343 Davis Hall
alphonce@buffalo.edu

Office hours:
Thursday 12:00 PM – 2:00 PM
Friday 8:30 AM – 10:30 AM

OR request appointment via e-mail
Turn off and put away electronics:

- cell phones
- pagers
- laptops
- tablets
- etc.
Last time

- association
- mutator/accessor methods

Today

- method calls in detail

Coming up

- interfaces
- event handling
channel 1
The course is moving...

A. too quickly
B. too slowly
C. just right
The labs have been ...

A. too easy
B. too difficult
C. just right
Are you learning?

A. I knew most of what we’ve from before.
B. I knew some things from before.
C. More everything is new to me.
REVIEW
public class Dog {
  private Collar _collar;
  private Sweater _sweater;
  private Tail _tail;
  public Dog(Collar c, Sweater s) {
    _collar = c;
    _sweater = s;
    _tail = new Tail();
  }
  public void setCollar(Collar abc) {
    _collar = abc;
  }
  public void setSweater(Sweater q) {
    _sweater = q;
  }
}

Similarities:
both set the value of an instance variable

Differences:
constructor sets value of an instance variable when the class is instantiated
mutator sets the value of an instance variable after the object already exists
constructor initializes ALL instance variables
mutator sets the value of just one instance variable
When method is called, body of method is carried out.

Local variables of method are allocated space in an **invocation record/stack frame**.

An invocation record is allocated space on the runtime stack.

*The invocation record is the entire context for the execution of the method*
Connection made!
Recall this code

package code;
public class Farm {

    private example1.BarnYard _t;

    public Farm() {
        _t = new example1.BarnYard();
    }

    public void addTwoChickens() {
        example1.Chicken c1 = new example1.Chicken();
        example1.Chicken c2 = new example1.Chicken();
        _t.addChicken(c1);
        _t.addChicken(c2);
        c1.start();
        c2.start();
    }
}

Which object’s instance variable does _t refer to?
package code;
public class Farm {

    private example1.BarnYard _t;

    public Farm() {
        this._t = new example1.BarnYard();
    }

    public void addTwoChickens() {
        example1.Chicken c1 = new example1.Chicken();
        example1.Chicken c2 = new example1.Chicken();
        this._t.addChicken(c1);
        this._t.addChicken(c2);
        c1.start();
        c2.start();
    }
}

Which object’s instance variable does _t refer to?
We discussed:
- invocation records
- ‘this’
- parameter and non-parameter local variables
- instance variables

We explored these via:
- memory diagrams
- object diagrams

We drew lots of diagrams on the whiteboard.