CSE115 / CSE503
Introduction to Computer Science I
Dr. Carl Alphonce
343 Davis Hall
alphonce@buffalo.edu

Office hours:
Tuesday 10:00 AM – 12:00 PM*
Wednesday 4:00 PM – 5:00 PM
Friday 11:00 AM – 12:00 PM
OR request appointment via e-mail

*Tuesday adjustments: 11:00 AM – 1:00 PM on 10/11, 11/1 and 12/6
ANNOUNCEMENTS
Scientista is having an event just for freshman on Monday, September 19 at 6:00 PM in Davis 113A.

We aim for this event to allow students to get to know each other, which we think is especially important during their first year. We also plan to talk about opportunities that they should take advantage of in their first year.
UB ROBOTICS

GENERAL MEETING
Bi Weekly Starting 09/16
Davis 230A at 6PM

PROJECTS
Battle Bots at Alfred State University
Intelligent Ground Vehicle Competition
First Person View Drone Racing

OTHER ACTIVITIES
Outreach Academia with Lego Mindstorms Workshops

WEBSITE: ENG.BUFFALO.EDU/UBR
FACEBOOK: FACEBOOK.COM/GROUPS/UB.ROBOTICS
ELECTRONICS:
off & away
Last time
- class definitions
- variables
- method calls
- object diagrams

Today
- Live demo
- class definitions in detail
- variables revisited

Coming up
- class relationships
REVIEW
Variables must be declared before use

- declaration specifies encoding scheme
- declaration specifies size

Declaration consists minimally of

- type (a class is a type)
- name

Examples

- example1.BarnYard by ;
- example1.Chicken c ;

The semicolon ';' is a terminator.
SYNTAX: \(<\text{variable}> = \text{<expression>};\)

‘=’ is the ASSIGNMENT OPERATOR (it is not ‘equals’!)

Example

\(\text{by} = \text{new example1.BarnYard();}\)

“\(\text{by is assigned the value of the expression ‘new example1.BarnYard()’}\)”

or

“\(\text{by is assigned a reference to a new example1.BarnYard() object}\)”

or

“\(\text{by is assigned a reference to a new BarnYard object}” (\text{example1 is implied})\)”
Developers write *class definitions*. placed in .java files

Compiler translate a .java file to a .class file.
e.g. Farm.java compiles to Farm.class

An object is an *instance of a class*.

Classes are instantiated only at runtime.
MOVING ON
object behaviors

to ‘send a message’ to an object we call a method on the object

sometimes we say ‘invoke a method’ rather than ‘call a method’
To put an example1.Chicken object inside an example1.BarnYard object, call the “addChicken” method of the example1.BarnYard object with a reference to an example1.Chicken object.

A method is called using a reference to the object on which we call the method.

> new example1.BarnYard().addChicken(new example1.Chicken())
new example1.BarnYard().addChicken(new example1.Chicken())
Dissecting a method call

```
new example1.BarnYard().addChicken(new example1.Chicken())
```

- An expression whose value is a reference to an object.
- The ‘member access operator’
- The name of the method.
- An argument list. In this example the list contains one expression.
Anatomy of a METHOD CALL

<expr> . <method> ( )

- A set of parentheses. Called an argument list. In this example the list is empty.
- The name of a method. The method must be defined for the type of object that the reference before the ‘.’ refers to.
- A period ‘.’, called ‘dot’ or the ‘member access operator’.
- An expression whose value is a reference to an object.
Anatomy of a METHOD CALL

\(<expr> \cdot <method>(<expr>)\)

- A set of parentheses with an expression inside. This is an argument list.
- The name of a method. The method must be defined for the type of object that the reference before the `.` refers to.
- A period `.` , called ‘dot’ or the ‘member access operator’.
- An expression whose value is a reference to an object.
Anatomy of a METHOD CALL

<expr> . <method> ( <expr>, <expr>,...,<expr> )

A period '.', called 'dot' or the 'member access operator'.

A set of parentheses containing comma-separated expressions. Also an argument list.

The name of a method. The method must be defined for the type of object that the reference before the ' .' refers to.

An expression whose value is a reference to an object.
Let us define a class which, when instantiated,
creates a BarnYard,
creates a Chicken,
adds the Chicken, and
makes the Chicken move

```java
example1.BarnYard by;
by = new example1.BarnYard();
example1.Chicken c;
c = new example1.Chicken();
by.addChicken(c);
c.start();
```

This is similar to what you will do for lab 2.
On to Eclipse for live coding demo!