University at Buffalo
The State University of New York

cse@buffalo

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*

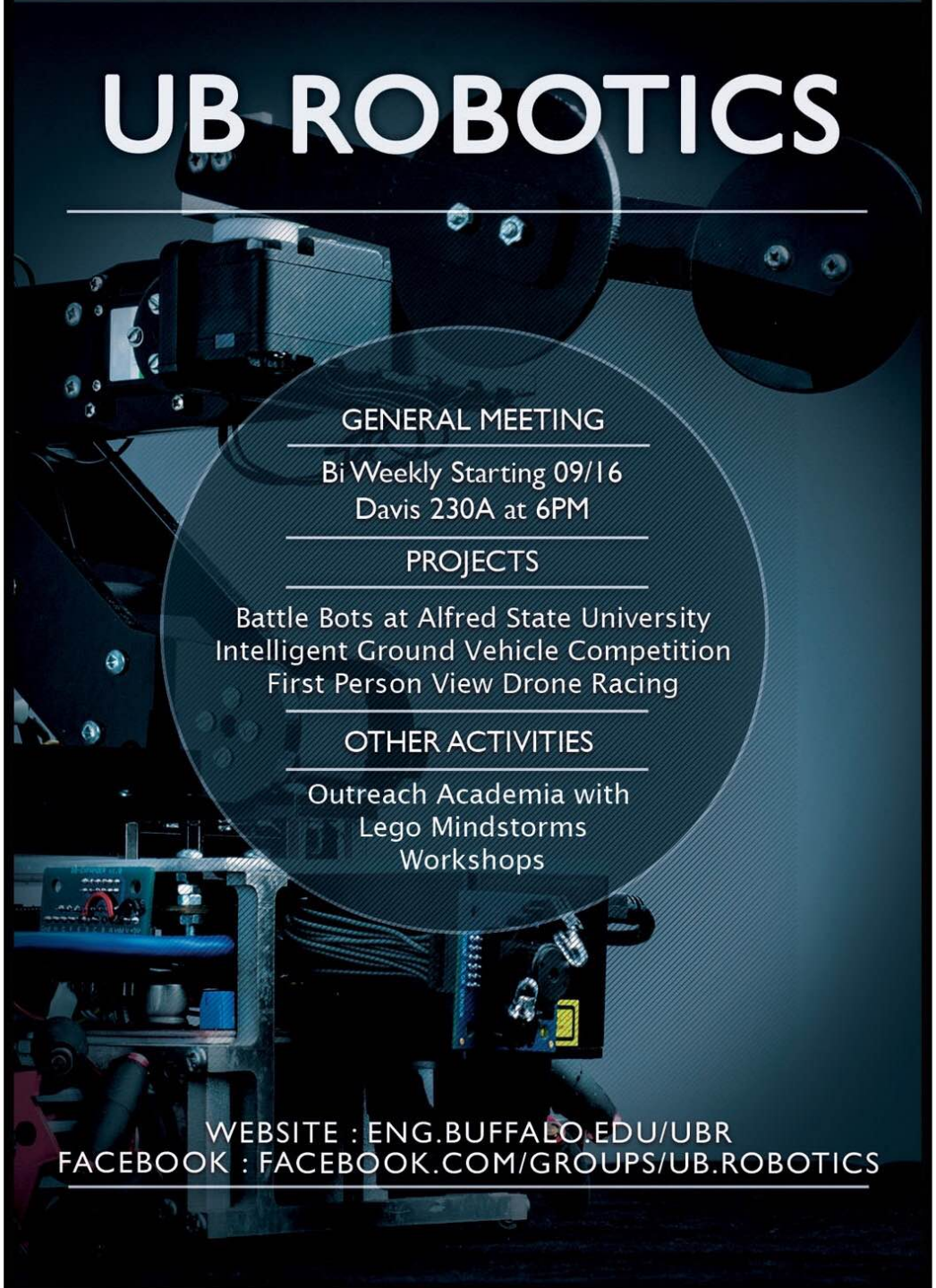
ANNOUNCEMENTS

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.

ANNOUNCEMENTS



ELECTRONICS:

off & away

ROADMAP

Last time

- class definitions
- variables
- method calls
- object diagrams

Today

- Live demo
- class definitions in detail
- variables revisted

Coming up

- class relationships

REVIEW

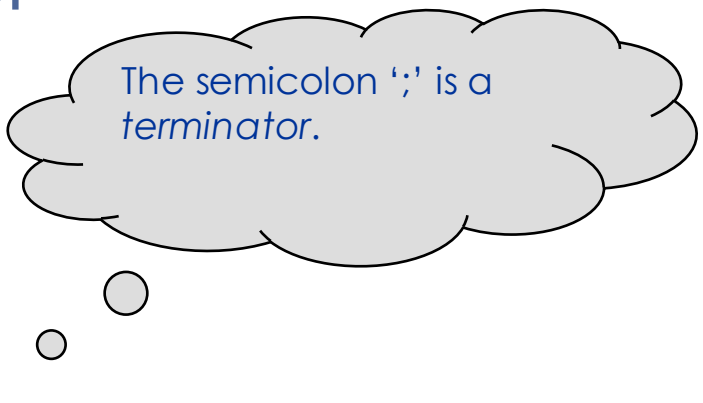
The variable declaration

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 ;
```

assignment statement

SYNTAX: *<variable> = <expression> ;*

'=' is the ASSIGNMENT OPERATOR (it is not 'equals'!)

Example

```
by = new example1.BarnYard();
```

"by is assigned the value of the expression 'new example1.BarnYard()' "

or

"by is assigned a reference to a new example1.BarnYard() object"

or

"by is assigned a reference to a new BarnYard object" (example1 is implied)

Classes vs. Objects

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

BACK TO OBJECTS

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*’

Object communication

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.

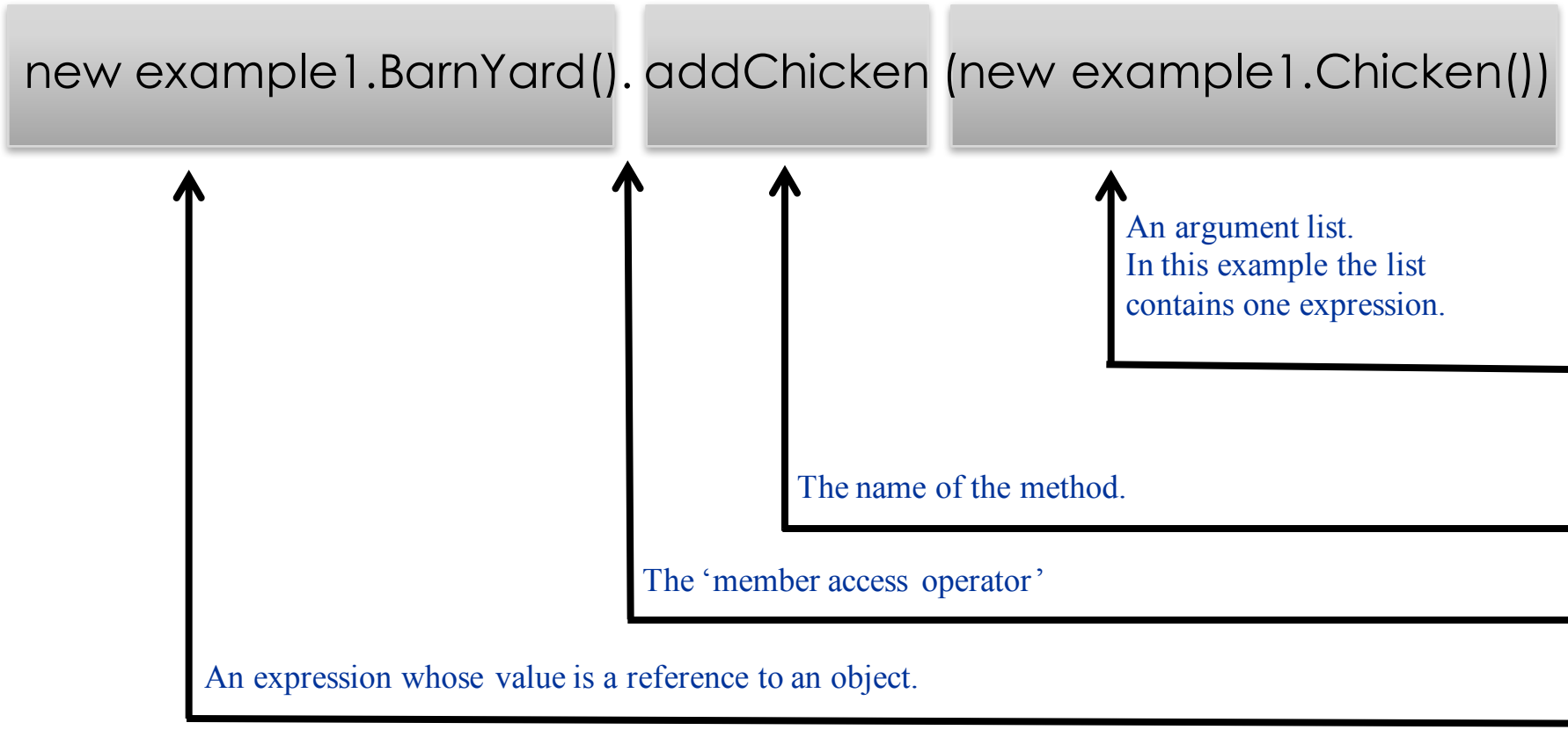
A method is called using a reference to the object on which we call the method.

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

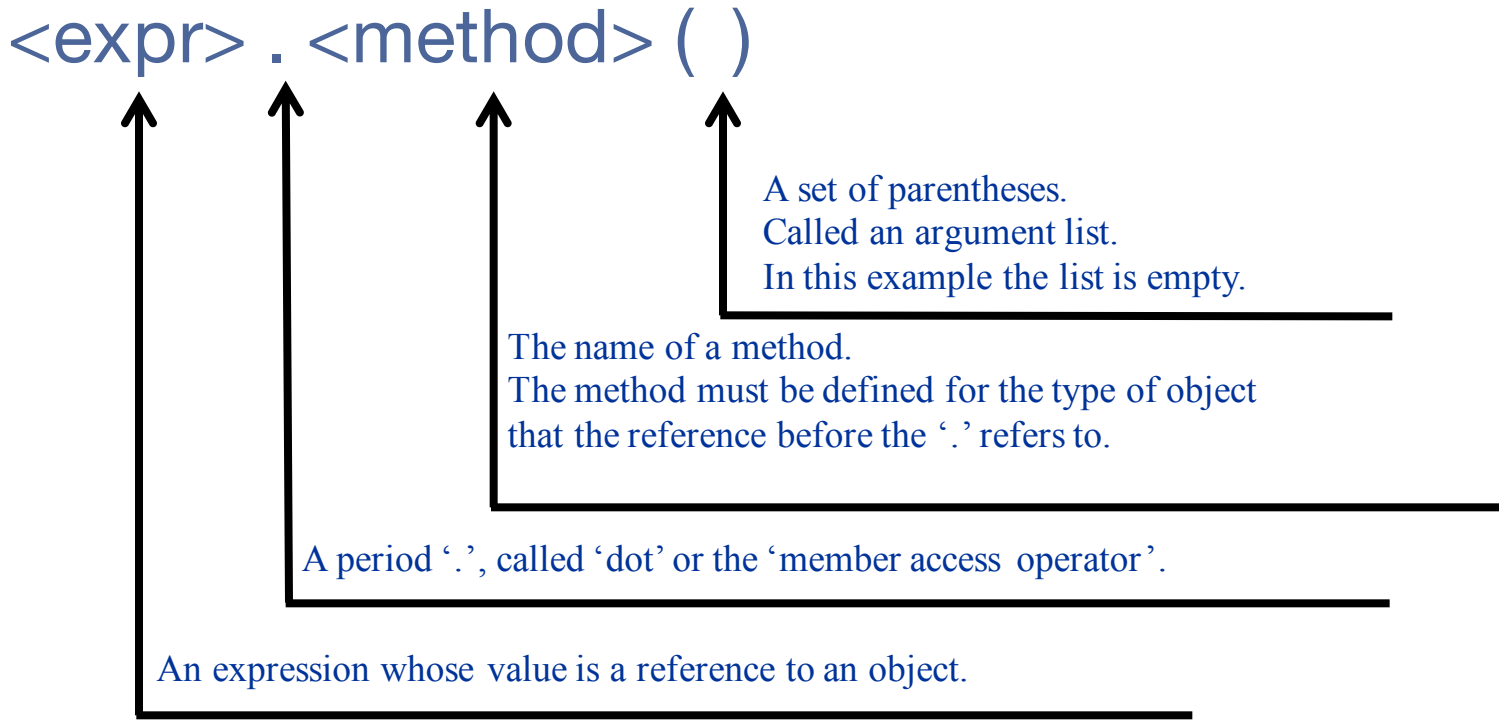
The method call from last time

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

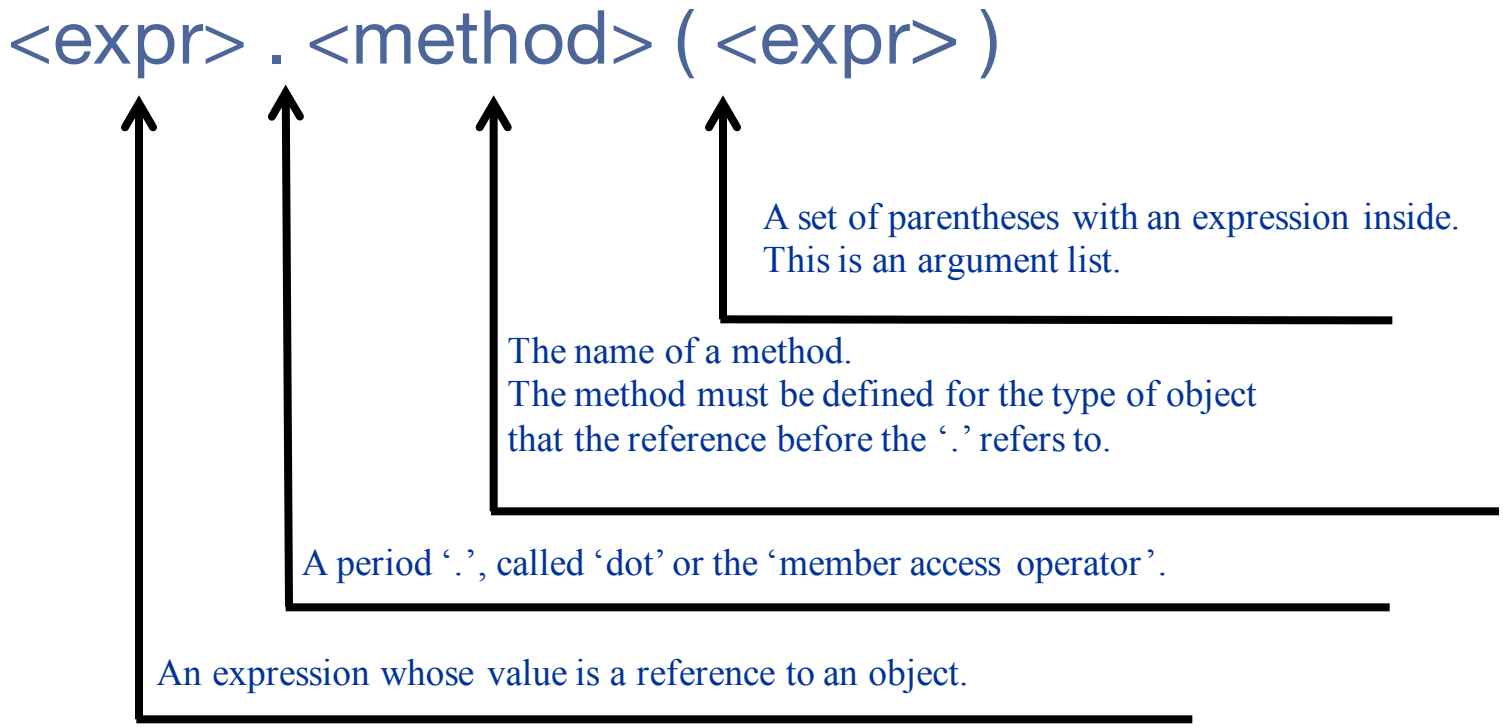
Dissecting a method call



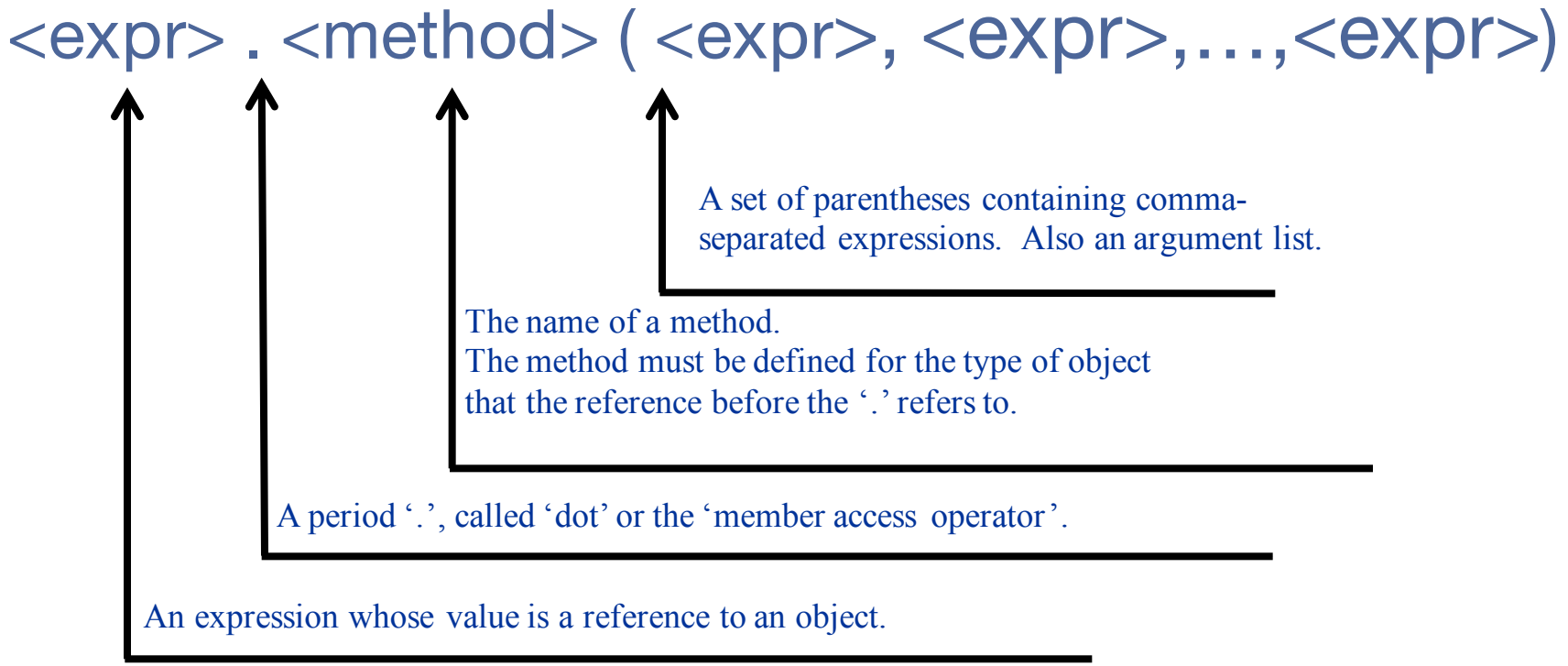
Anatomy of a METHOD CALL



Anatomy of a METHOD CALL



Anatomy of a METHOD CALL



Defining a class

Let us define a class which, when instantiated,

creates a BarnYard,
creates a Chicken,
adds the Chicken, and
makes the Chicken move

```
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.

DEMO

On to Eclipse for live coding demo!