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*

ROADMAP

Last time

Control structures (repetition)

Coding exercise

Today

Collections

Coding exercises

Coming up

Iterators

EXERCISE REVIEW

PAIR CODING EXERCISE

Define a class quiz.Question. In this class define a method named answer.

Define this method so that it returns a String consisting of all the integers from 0 to n inclusive, comma-separated, if n >= 0, and the String "0" otherwise. For example:

> **answer(-1) must return "0"**
> **answer(0) must return "0"**
> **answer(3) must return "0, 1, 2, 3"**

Submit to Exercise-03 in Web-CAT – enter the usernames of everyone in your group!

You have 20 minutes to complete this exercise.

Exercise-03

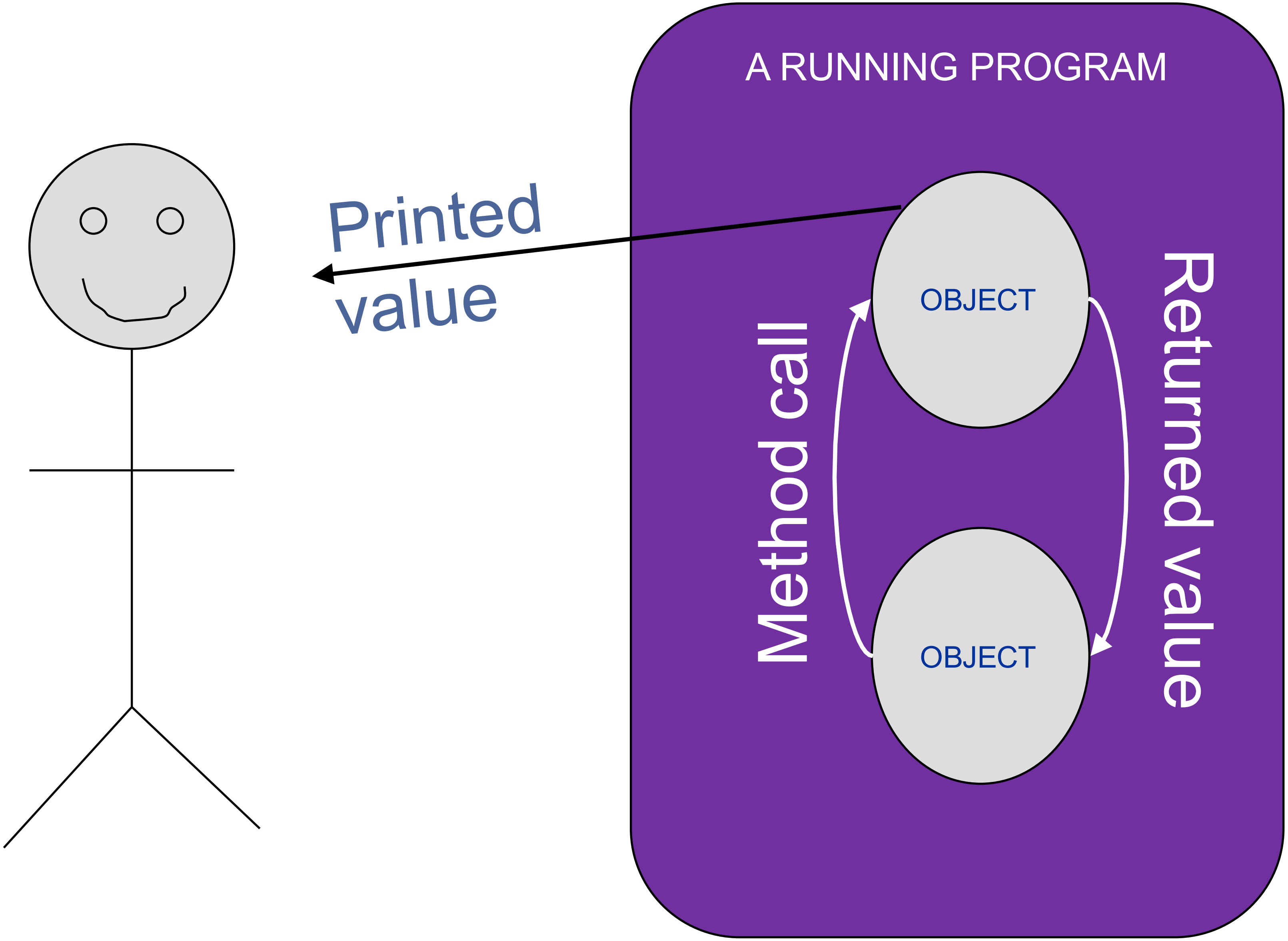
```
package quiz;

public class Question {

    public String answer(int n) {
        String answer = "0";
        int i = 1;
        while (i <= n) {
            answer = answer + ", " + i;
            i = i + 1;
        }
        return answer;
    }

}
```

returned value vs. printing



MOVING ON

COLLECTIONS

Collections

A collection object can store arbitrarily many (references to) objects.

We will first learn to become users/clients of collections.

Next semester you will learn to become builders of collections.

java.util.Collection<E> interface

All collection classes in Java are subtypes of the java.util.Collection<E> interface.

‘<E>’ is new syntax

E is a *type variable*, and denote the element type of the collection:

Collection<String> denotes a collection of String objects

Collection<ActionListener> denotes a collection of ActionListener objects

java.util.Collection<E> interface

Among the methods specified in the interface:

boolean add(E item) --- tries to add item to the collection; if this is successful, true is returned, false otherwise

boolean remove(Object item) --- tries to remove (one occurrence of) item from the collection; if this is successful, true is returned, false otherwise

boolean contains(Object item) --- returns true if item is in the collection, false otherwise

int size() --- return the number of items currently in the collection

Two specific collections
(defined in java.util)

ArrayList<E>

- permits duplicates
- allows client to control order of elements

HashSet<E>

- does not permit duplicates
- does not allow client to control order of elements

Collections: usage example

To declare a variable of type HashSet of String:

```
HashSet<String> names;
```

To create a HashSet of String object, and assign its reference to the variable declared above:

```
names = new HashSet<String>();
```

On to eclipse!

(examples from lecture are in the LectureCode project in class repo)