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*

ROADMAP

Last time

- Control structures (selection)
- Coding exercise

Today

- Control structures (repetition)
- Coding exercises

Coming up

- Collections

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 "NEGATIVE" if its int argument is negative, "POSITIVE" if its argument is positive, and "ZERO" otherwise.

Submit to Exercise-02 in Web-CAT – enter the usernames of everyone in your group!

You have 15 minutes to complete this exercise.

Exercise-01

```
package quiz;
```

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

Define this method so that it returns "NEGATIVE" if its int argument is negative, "POSITIVE" if its argument is positive, and "ZERO" otherwise.

Exercise-01

```
package quiz;

public class Question {



}
```

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

Define this method so that it returns "NEGATIVE" if its int argument is negative, "POSITIVE" if its argument is positive, and "ZERO" otherwise.

Exercise-01

```
package quiz;

public class Question {

    public          answer(      ) {




    }

}
```

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

Define this method so that it returns "NEGATIVE" if its int argument is negative, "POSITIVE" if its argument is positive, and "ZERO" otherwise.

Exercise-01

```
package quiz;

public class Question {

    public String answer(      ) {

    }

}
```

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

Define this method so that it returns "NEGATIVE" if its int argument is negative, "POSITIVE" if its argument is positive, and "ZERO" otherwise.

Exercise-01

```
package quiz;

public class Question {

    public String answer(int x) {

    }

}
```

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

Define this method so that it returns "NEGATIVE" if its int argument is negative, "POSITIVE" if its argument is positive, and "ZERO" otherwise.

Exercise-01

```
package quiz;

public class Question {

    public String answer(int x) {
        if (x < 0) {
            return "NEGATIVE";
        }

    }

}
```

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

Define this method so that it returns "NEGATIVE" if its int argument is negative, "POSITIVE" if its argument is positive, and "ZERO" otherwise.

Exercise-01

```
package quiz;

public class Question {

    public String answer(int x) {
        if (x < 0) {
            return "NEGATIVE";
        }
        else if (x > 0) {
            return "POSITIVE";
        }


    }

}
```

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

Define this method so that it returns "NEGATIVE" if its int argument is negative, "POSITIVE" if its argument is positive, and "ZERO" otherwise.

Exercise-01

```
package quiz;

public class Question {

    public String answer(int x) {
        if (x < 0) {
            return "NEGATIVE";
        }
        else if (x > 0) {
            return "POSITIVE";
        }
        else {
            return "ZERO";
        }
    }

}
```

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

Define this method so that it returns "NEGATIVE" if its int argument is negative, "POSITIVE" if its argument is positive, and "ZERO" otherwise.

MOVING ON

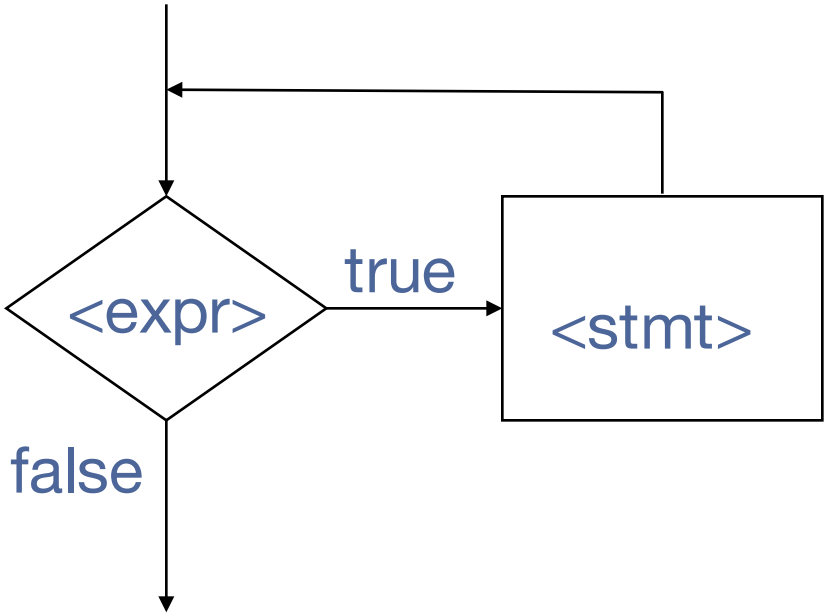
Control Structures

→ repetition

→ while statement

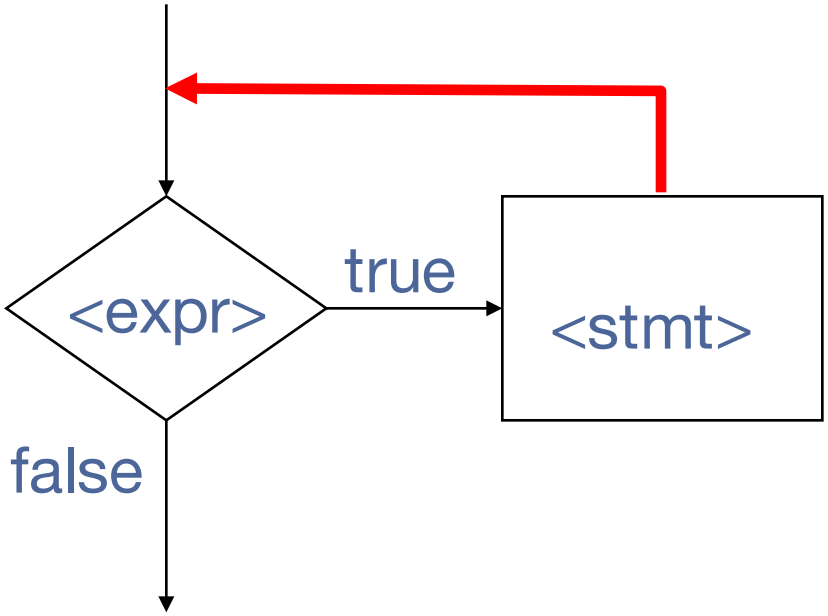
while statement

```
while ( <expr> ) <stmt>
```



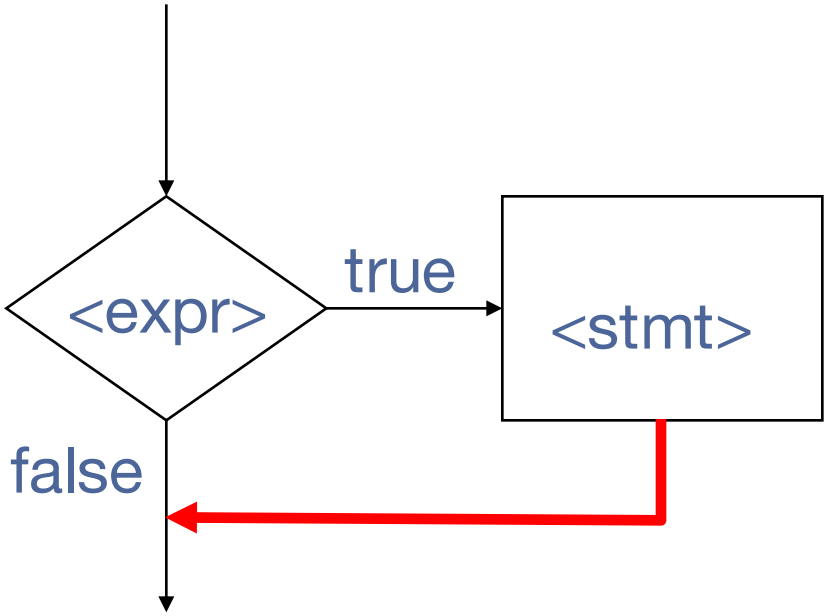
while statement

```
while ( <expr> ) <stmt>
```

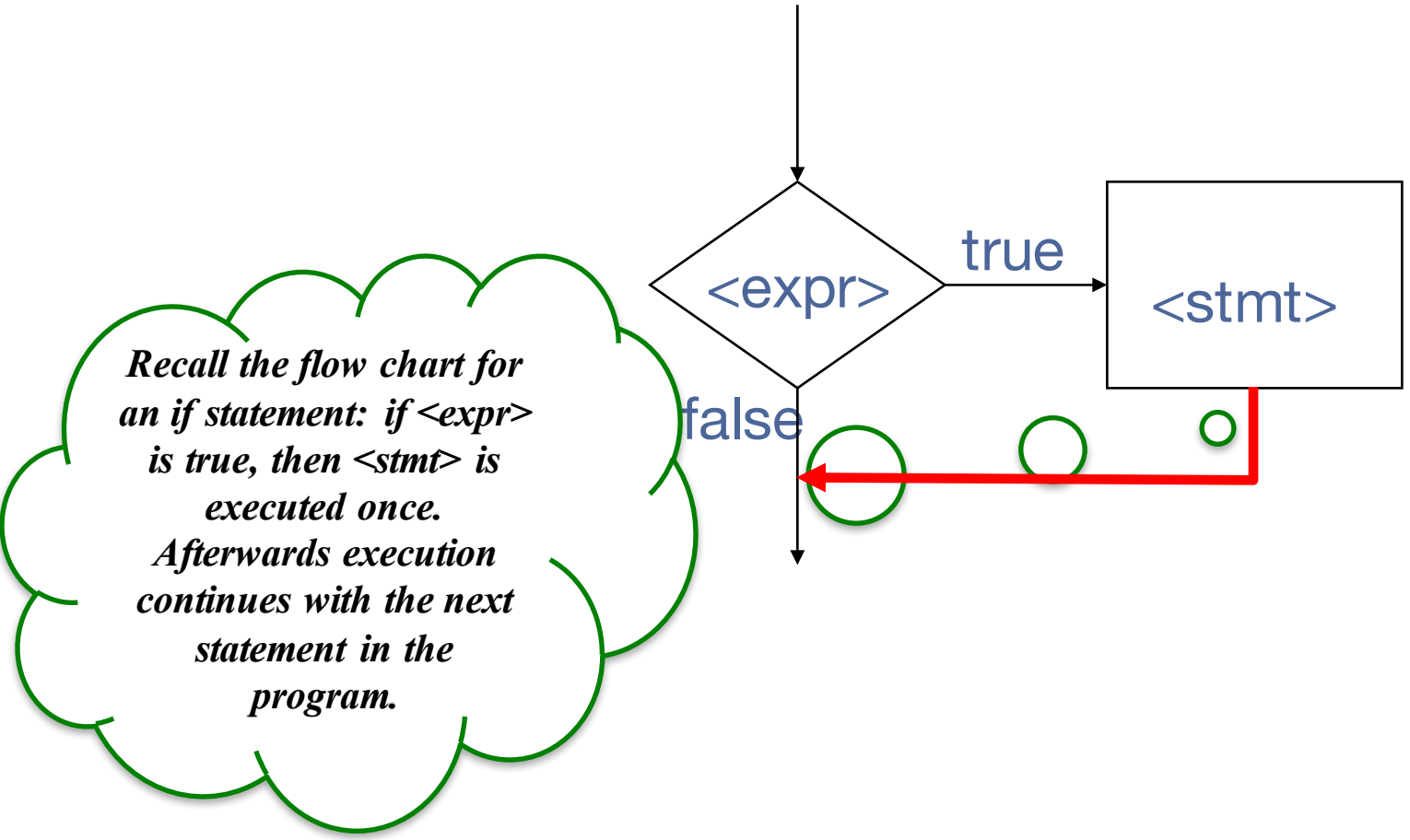


if statement

```
if ( <expr> ) <stmt>
```

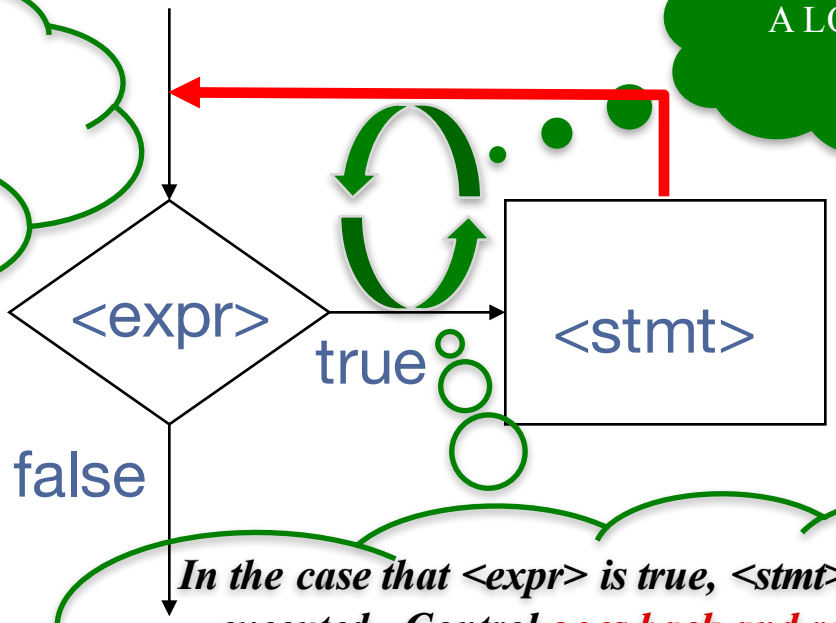


```
if ( <expr> ) <stmt>
```


<expr>
true
<stmt>
false
Recall the flow chart for an if statement: if <expr> is true, then <stmt> is executed once. Afterwards execution continues with the next statement in the program.

`while ( <expr> ) <stmt>`

*The flow chart for a **while statement** is different from that of the **if statement** in one very important respect.*



*In the case that <expr> is true, <stmt> is executed. Control **goes back and re-evaluates <expr>, thus creating a loop in the flow of control**. In the case that <expr> is false the loop is not entered and program execution continues with the next statement.*

while loop example

```
public void message() {

    System.out.print("I ");

    int timesAlreadyPrinted = 0;

    while (timesAlreadyPrinted < 3) {
        System.out.print("really ");
        timesAlreadyPrinted = timesAlreadyPrinted + 1;
    }

    System.out.println(" like spring break!");
}
```

while loop example

```
public void message(int numberOfRepetitions) {

    System.out.print("I ");

    int timesAlreadyPrinted = 0;

    while (timesAlreadyPrinted < numberOfRepetitions){
        System.out.print("really ");
        timesAlreadyPrinted = timesAlreadyPrinted + 1;
    }

    System.out.println(" like spring break!");
}
```

tracing code

We spent some time tracing the execution of the code on the previous slide, keeping track of the value of the variable *timesAlreadyPrinted* and the corresponding output.

Being able to trace the execution of code by hand is an important skill.

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.