

# REPETITION: WHILE LOOPS

Deborah A. Trytten  
CS 1323/1324

# REVIEW AND PREVIEW

## ○ What we've learned so far:

- Data types
- Variables
- Numerical operations
- Logical operations
- Input and output
- Conditional statements

## ○ What we're going to learn next!

- Loops (while and for)
- Methods
- Arrays

# WHAT IF WE WANT TO...

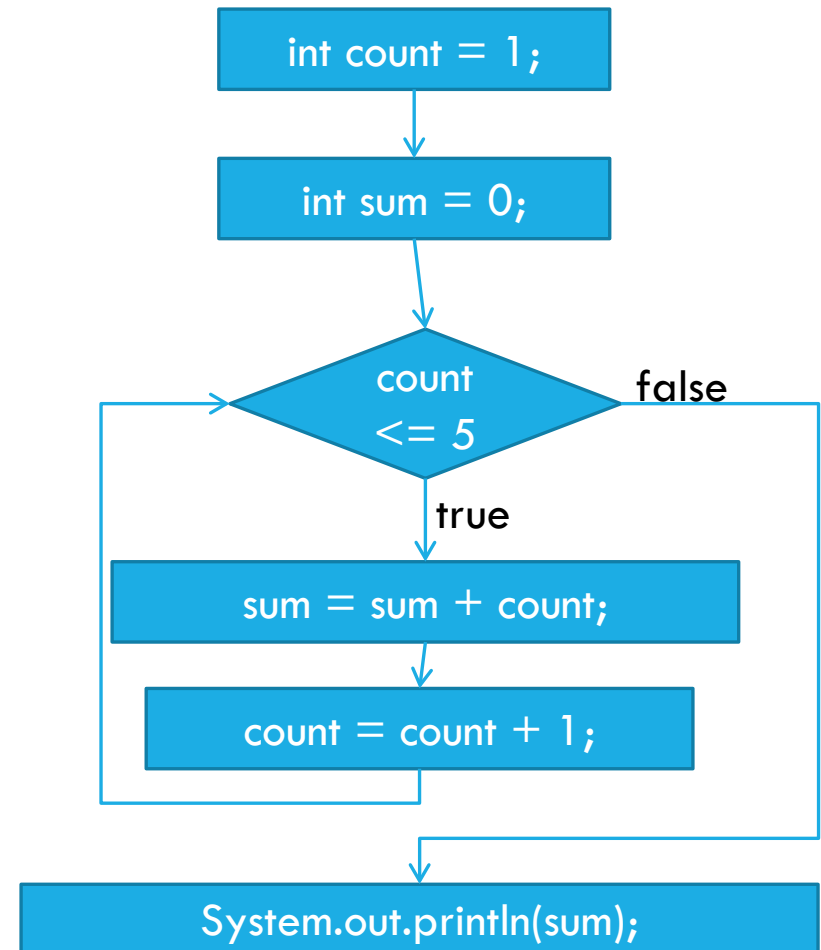
- Allow a user to purchase as many items as they wish from our website
- Find the average of 10,000 numbers entered from the keyboard
- Need to be able to perform repetitive operations
  - Count yes responses from the user (Project 4)
- Loops!
  - Another way of changing the order in which statements execute
  - Like if/else

# CONTROL FLOW

- Sum 1 to 5

```
int count = 1;
int sum = 0;
while (count <= 5)
{
    sum = sum + count;
    count = count + 1;
}
System.out.println(sum);
```

- What is count at end?



# TRACING CODE (DEBUGGING)

- Keep track of the contents of variables in a table
  - Just like tracing conditional statements
  - Reminder: like a memory diagram, but without crossing out
  - Reminder: bottom value in table is current value

```
int sum = 0;
int counter = 3;
while (sum < 23)
{
    sum = sum + counter;
    counter = counter + 2;
}
```

# OBSERVATION: WHILE

- The loop body does not have to be executed

```
int size = 9;
while (size < 7)
{
    size = size - 2;
}
```

# THINK PAIR SHARE

- Trace the loop below

```
int counter = 1;
int sum = 0;
while (counter < 7)
{
    counter = counter + 8;
    sum = sum + counter;
    counter = counter - 5;
}
```

# IClicker QUESTION

- What is the value of sum at the end of this loop?

```
int sum = 0;
while (sum < 23)
{
    sum = sum + 5;
}
```

a: 20

b: 23

c: 25

d: 30



# COUNTING LOOPS

- Write a loop that finds the average of five numbers entered from the keyboard
- Two possible methods:
  - Start with 0, end  $< 5$
  - Start with 1, end  $\leq 5$
- What is the advantage of starting with 0?
- Using these patterns avoids a lot of mistakes

# ANNOUNCEMENTS AND REMINDERS

- TC 6 due Sunday night
  - Asterisks problem: add newline character after the asterisks
- Ch. 8 due Sunday night
  - Starting methods next week

# OBOB ANTIBUGGING TECHNIQUE

- Modifying the loop bounds for three iterations seems to work well
  - Less and you don't have start, middle, end
  - More and you get bored and lose count
- Example: Find the smallest of ten values entered from the keyboard

# IClicker QUESTION

- How many times is the loop body below executed?

```
int count = 0;
while (count <= 5)
{
    count = count + 1;
}
```

a: 4

b: 5

c: 6

d: 7

# INFINITE LOOPS

```
int sum = 0;
int count = 1;
while (count <= 5)
{
    sum = sum + count;
}
```

- What's the problem?
- What does this look like in eclipse?
  - Stop with little red square over console

# TECHNIQUE: SENTINEL CONTROL

- Users don't like having to tell you how many values you're going to process in advance
  - Easier for them to let computer count
  - Much more accurate
- Users will give value that indicates the end of values
  - Called a sentinel
  - Choosing this value can be problematic—can't be legal data

# EXAMPLE

- Example: Find the average of student grades in the class
  - Find a good sentinel
  - Write the code (without priming read)
- Technique: Putting a conditional inside of a loop
  - Nesting

# IClicker Question

Suppose you are reading in people's first names at the keyboard. Which of the values below would not be a good sentinel?

- a) Fred
- b) -1
- c) Quit
- d) End



# PRIMING READS

- Sentinel controlled loops usually have priming reads
  - Read data before loop starts
  - Read data for the next iteration at the end of the previous iteration
- Used to make sure that the sentinel doesn't get used as data
- Why?

# IClicker Question

- Does the code below use a priming read?

```
int target; // value assigned elsewhere
int data = -1;
while (data != 0)
{
    data = input.nextInt();
    if (data == target)
        System.out.println("Target found");
}
```

a: True

b: False

# DECREMENTING LOOPS

- Loop counters can decrement as well as increment
- Sometimes it is much easier to write a loop with a decrement rather than increment
  - Remember this as a possibility
  - Especially when you're having trouble figuring out how to write a loop

# EXAMPLE

- Write a loop that finds the largest factor of a number (the number itself is excluded)
  - Use decrement
- Why is it better to do this as a decrement?

# IClicker Question

- Which of the tables below correctly traces this loop?

```
int x = 0;
int y = 10;
while (x < y) {
    x = x + 2;
    y = y - 1;
}
```

a:

x	y
0	10
2	9
4	8

b:

x	y
0	10
2	9
4	8
6	7

c:

x	y
0	10
2	9
4	8
6	7
8	6

d: None correct

# EXAMPLE PROGRAM

- Write a program that counts how many students passed (70 and over) or failed (under 70) an examination
  - Enter data at the command line, separated by spaces