For Loops

CS 1323 and CS 1324

Deborah A Trytten

Example: Count Input Values

- Count the number of positive values entered before a negative value is entered:

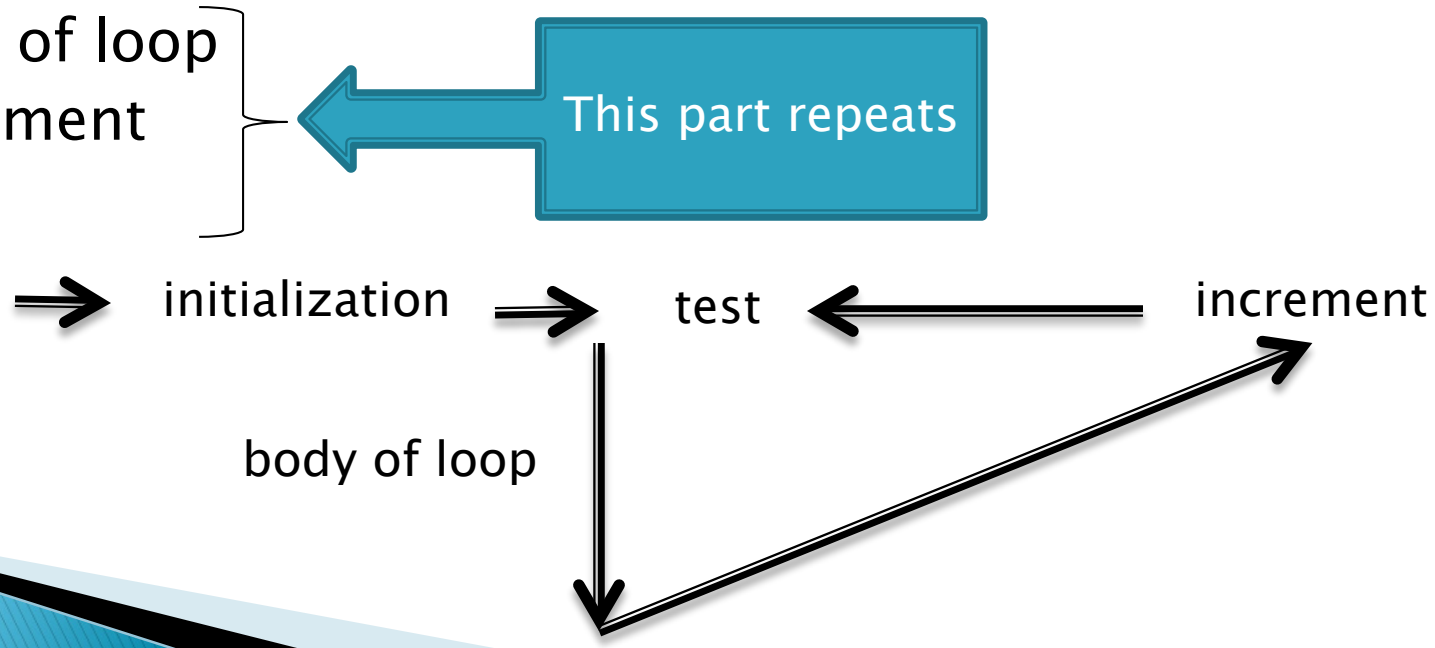
```
int value = keyboard.nextInt(); // priming read
int count;
for (count = 0; value > 0; count = count + 1)
{
    value = keyboard.nextInt();
}
```

- Initialization and test occur before first iteration
  - Just like while loop
- Variables declared in header are active only during loop

Order of Execution

- Pattern
  - for (initialization; test; increment)

- Order of execution
  - Initialization
  - Test
  - Body of loop
  - Increment
  - Test



Example: Reverse String

- Write a method using a for loop that creates a new String that has the characters in a given String reversed

- Example “abcde” would create “edcba”
  - Use charAt and length methods in String class

Think Pair Share

- Write a for loop that prints out each character in a string on a separate line.

- Example: if the string is “ABC”, the loop should print
  A
  B
  C

- Assume that a variable named “source” stores a reference to the string:
  - `String source = "ABC";`

iClicker Question

- How many times will the loop shown below execute?

```
int count = 0;
for (int num = 0; count > 0; num = num + 1) {
    if (num % 2 == 0) {
        count = count + 1;
    }
}
```

a) 0 b) infinite c) 1 d) 10

Shortcuts

- Auto increment and decrement operators
  - `++i  // increment then use (usually better)`
  - `i++  // use then increment`

```
for (int i = 0; ++i < 3; ) {  // Trace ++i and i++
    System.out.println(i);
}
```

- Comma can separate multiple initialization (of same data type) and increment/decrement statements:

```
for (int x = 0, y = 1; x < 10; ++x, --y) {
    if (x % 2 == 0) {
        ++y;
    }
}
```

iClicker Question

- True or false: if we replace ++i with i++ below, the loop will do the same thing?

```
for (int i = 0; i < 10; ++i)
{
    System.out.println(i);
}
```

a) True b) False

Empty Areas in For Loop Header

- Any of the three areas can be empty
  - Even all three
  - Semicolons are required

- Usually a sign that a while loop is a better choice

```
int j = 0;
for (; j < 5;) {
    ++j;
}
```

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

- When the test is empty, it is evaluated as true
  - The following are all equivalent:

```
for (;;)    for (; true;)    while (true)
```

Example: Get User Input

- Write a for loop that reads user input until an integer is found in a given range
- Assume the variable keyboard is a Scanner that is declared and initialized
- Rewrite this with a while loop

Choosing While or For Loop

- You can accomplish any repetition with either a while or for loop.

- However…
  - Loops that are controlled with a counter are naturally for loops.
  - Loops without an initialization and an increment are naturally while loops.

Common Errors

- Extraneous semicolon:

```
for (int j = 0; j < 10; ++j);
    System.out.println(j);
```

- Trying to use a variable declared in the for-loop header outside of the loop body:

```
for (int x = 0, y = 1; x < 5; ++x)
{
    if (x % 2 == 0)
        y = y + 1;
}
// Can't use variables x or y here!
```