I NEED TO USE THE COMPUTER.
HOLD ON. LET ME SEE IF ANYONE WROTE ON MY FACEBOOK WALL... NOPE.
NOW LET ME SEE IF ANYONE REPLIED TO ME ON TWITTER... NOPE.
NOW LET ME SEE IF ANYONE STARTED FOLLOWING MY TUMBLR... NOPE.
DONE?
LET ME CHECK FACEBOOK AGAIN. SOMEONE COULD HAVE WRITTEN SOMETHING IN THE LAST 30 SECONDS... NOPE.
www.foxtrot.com twitter: @billamend
NOW LET ME CHECK TWITTER AGAIN... NOW TUMBLR...
©2011 Bill Amend / Dist. by Universal Uclick
OUR SISTER HAS GONE INFINITELY LOOPY.
NOW FACEBOOK... NOW TWITTER... NOW TUMBLR... NOW FACEBOOK... NOW TWITTER... NOW TUMBLR... NOW FACEBOOK... NOW TWITTER... NOW TUMBLR...
AMEND 4-10

Based on slides by Chand John
Used with permission
https://www.cs.utexas.edu/~chand/cs312/

Some text from slides by Larry Baker for IB Computer Science

Repetition with `for` loops

- So far, repeating a statement is redundant:

```
System.out.println("Mike says:");
System.out.println("Do Practice-It problems!");
System.out.println("Do Practice-It problems!");
System.out.println("Do Practice-It problems!");
System.out.println("Do Practice-It problems!");
System.out.println("Do Practice-It problems!");
System.out.println("It makes a HUGE difference.");
```

- Java's **`for` loop** statement performs a task many times.

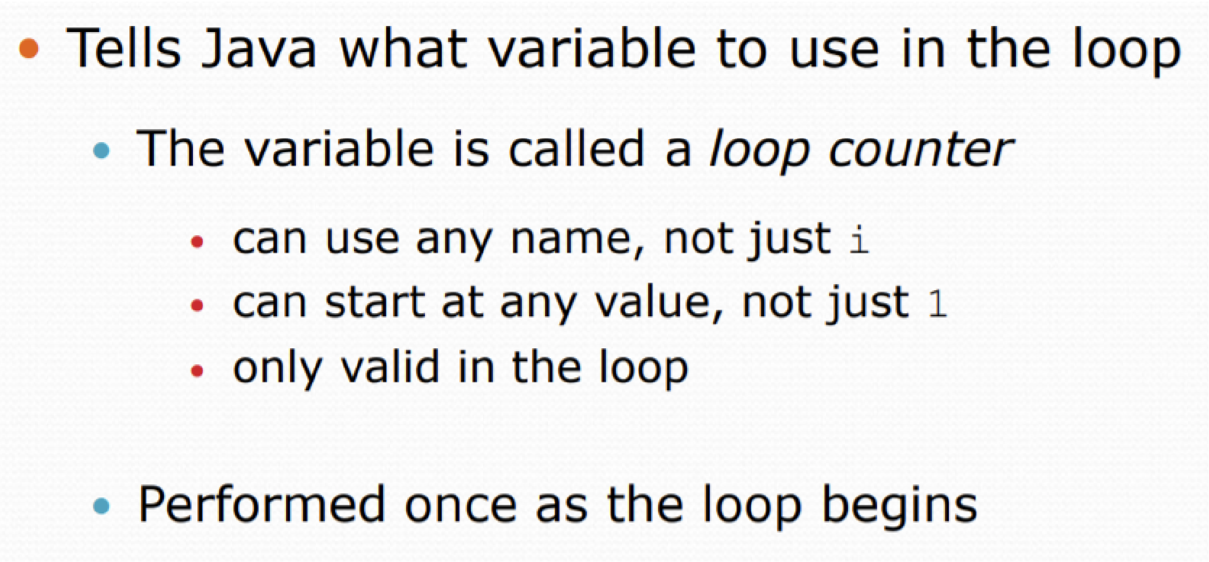
```
System.out.println("Mike says:");
for (int i = 1; i <= 5; i++) {   // repeat 5 times
    System.out.println("Do Pratice-It problems!");
}
System.out.println("It makes a HUGE difference.");
```

It is essential that a program be able to execute the same set of instructions many times: otherwise a computer would do only as much work as a programmer!

Initialization

Loop counter

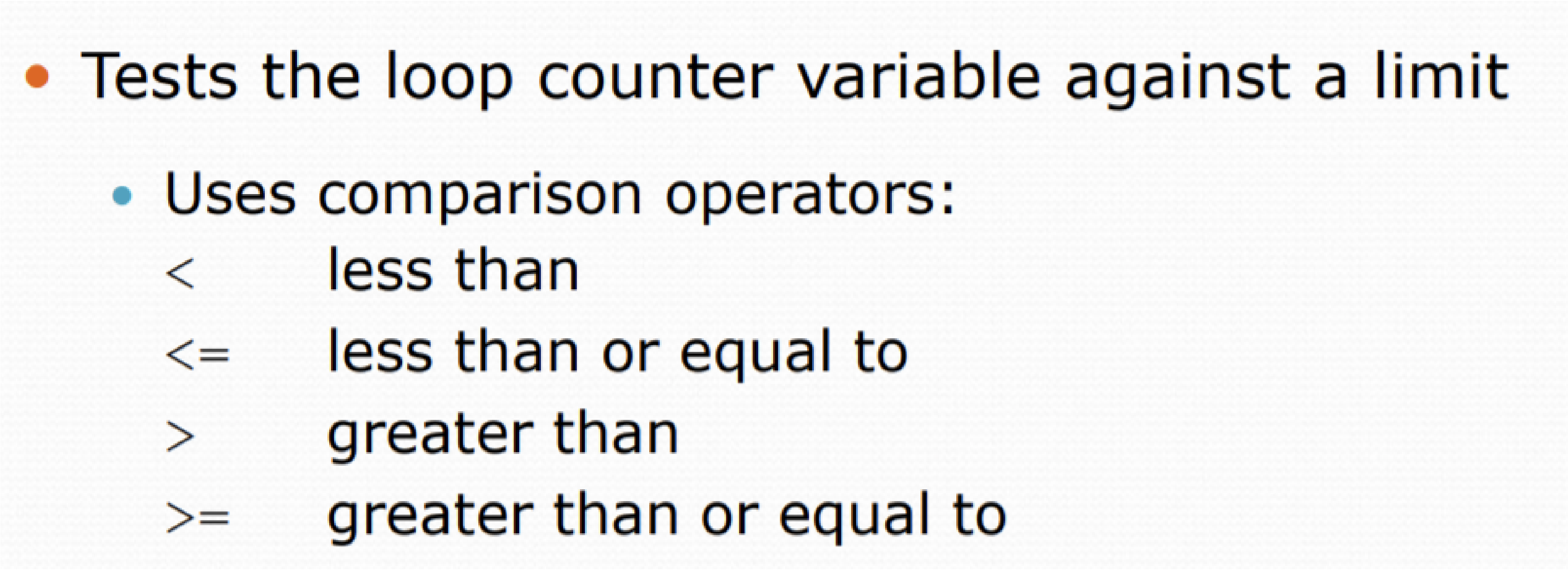
```
for (int i = 1; i <= 5; i++) {
    System.out.println("Do Practice-It!");
}
```



- Tells Java what variable to use in the loop
  - The variable is called a *loop counter*
    - can use any name, not just `i`
    - can start at any value, not just `1`
    - only valid in the loop
  - Performed once as the loop begins

Test

```
for (int i = 1; i <= 5; i++) {
    System.out.println("Do Practice-It!");
}
```



- Tests the loop counter variable against a limit
  - Uses comparison operators:

<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to

Update

```
for(int i = 1; i <= 5; i++) {
    System.out.println("Do Practice-It!");
}
```

Aside: Increment and Decrement Operators

```
int x = 2;
x++;                    // x = x + 1;
                        // x now stores 3

double gpa = 2.5;
gpa--;                  // gpa = gpa - 1;
                        // gpa now stores 1.5
```

Aside: Modify-and-assign operators

shortcuts to modify a variable's value

```
x += 3;                 // x = x + 3;
gpa -= 0.5;             // gpa = gpa - 0.5;
number *= 2 + 1;        // number = number * (2 + 1);
```

Clicker

- What is output by the following code?

```
int x = 2;
int y = 5;
x *= 3 + y + x;
System.out.println(x + " " + y);
```

A. **20 5**

B. **2 5**

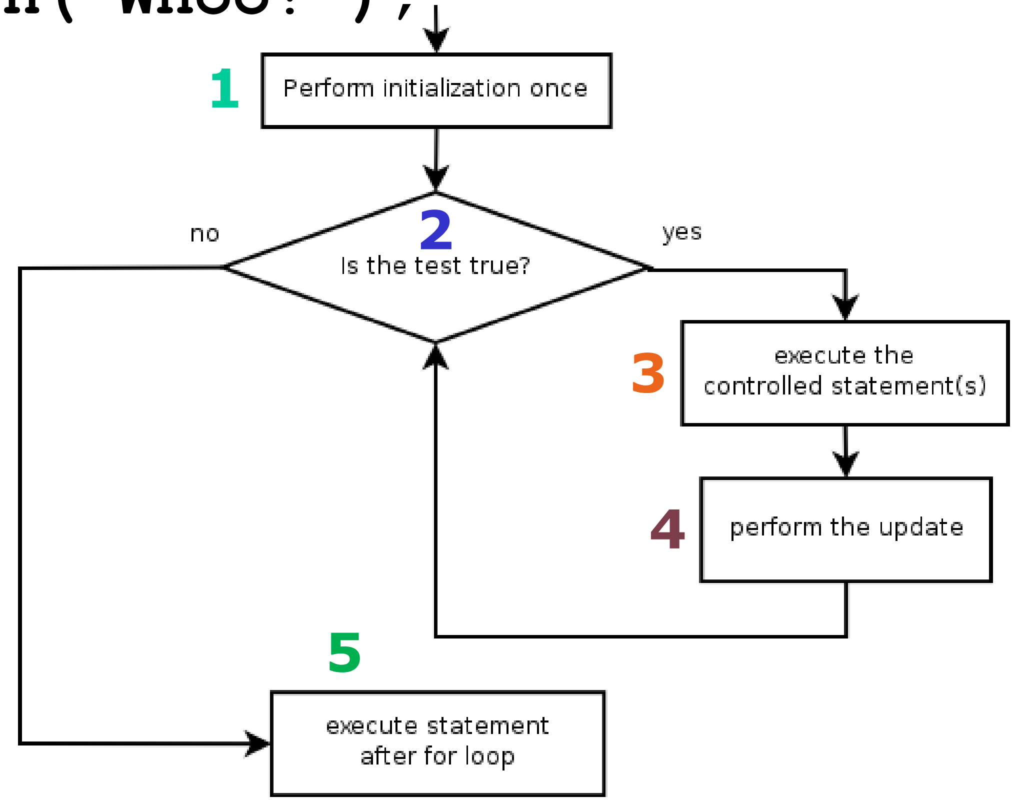
C. **13 5**

D. **20 10**

E. Something other than A - D

Loop walkthrough

```
           1      2     4
for (int i = 1; i <= 4; i++) {
  3 System.out.println(i + " squared = " + (i * i));
}
5 System.out.println("Whoo!");
```



1
Perform initialization once
no
2
Is the test true?
yes
3
execute the controlled statement(s)
4
perform the update
5
execute statement after for loop

Output:

```
1 squared = 1
2 squared = 4
3 squared = 9
4 squared = 16
Whoo!
```

System.out.print

- Prints without moving to a new line

```
System.out.print(“Hey, ”);
System.out.println(“how’s it going?”);
```

- Output:

```
Hey, how’s it going?
```

Clicker Question

- How many asterisks are output by the following code?

```
for(int i = -2; i <= 13; i++) {
    System.out.print("*");
    System.out.print("**");
}
```

A. 0

B. 15

C. 45

D. 48

E. 68

Counting down

```
System.out.print("T-minus ");
for (int i = 10; i >= 1; i--) {
    System.out.print(i + ", ");
}
System.out.println("blastoff!");
System.out.println("The end.");
```

Output:

```
T-minus 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, blastoff!
The end.
```

Nested loops

reading: 2.3

Nested loops

- **nested loop**: A loop placed inside another loop.

```
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 10; j++) {
        System.out.print("*");
    }
    System.out.println();   // to end the line
}
```

- Output:

```
**********
**********
**********
**********
**********
```

- The outer loop repeats 5 times; the inner one 10 times.
  - "sets and reps" exercise analogy

Nested `for` loop exercise

- What is the output of the following nested `for` loops?

```
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= i; j++) {
        System.out.print("*");
    }
    System.out.println();
}
```

- Output:

```
*
**
***
****
*****
```

clicker Question

- What is output by the following code?

```
int total = 0;
for(int i = 1; i <= 4; i++) {
    for(int j = 1; j <= i; j++) {
        total += i;
    }
}
System.out.println(total);
```

A. 10 B. 20 C. 30 D. 40 E. 50

Loop tables

- To see patterns, make a table of `count` and the numbers.

`count`	number to print	`5 * count`	`5 * count - 3`
1	2	5	2
2	7	10	7
3	12	15	12
4	17	20	17
5	22	25	22

Common errors

- Both of the following sets of code produce *infinite loops*:

```
for (int i = 1; i <= 5; i++) {
    for (int j = 1; i <= 10; j++) {
        System.out.print("*");
    }
    System.out.println();
}

for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 10; i++) {
        System.out.print("*");
    }
    System.out.println();
}
```