COMPSCI 121: LOOPS II

Spring 20

GOALS FOR TODAY'S CLASS

Project 3 - Overview

More nuts & bolts of programming!

- **Looping part II:**
  - More with Looping and Strings.
  - Break and Continue
  - Nested loops.

PROJECT 3 - OVERVIEW - DEMO

Random

```
public Random()
```

Creates a new random number generator.

```
public Random(long seed)
```

Creates a new random number generator using a single `long` seed.

Demo:
Testing from Main
Calling a helper method.

REMINDER: HOW TO WRITE A LOOP

1. Determine purpose of loop
   a. Pick a loop structure (`while, for, do_while`)
2. Determine
   a. start condition
   b. termination condition
   c. update condition
3. Write loop body
   a. Determine body that is repeated
   b. Update loop control variable to reach termination

For every iteration of loop, write down values.

REVIEW: WRITING LOOPS -1

PROBLEM: Find the average of a set of positive integers entered by the user.

SOLUTION: Pseudocode.

```
Let sum = 0      // The sum of the integers entered by the user.
Let count = 0    // The number of integers entered by the user.
while there are more integers to process:
    Read an integer
    Add it to the sum
    Count it
Divide sum by count to get the average
Print out the average
```

What is the problem?

REVIEW: WRITING LOOPS - 2

PROBLEM: Find the average of a set of positive integers entered by the user.

SOLUTION: We test whether there are more integers to process.

```
Let sum = 0      // The sum of the integers entered by the user.
Let count = 0    // The number of integers entered by the user.
while there are more integers to process:
    Read an integer
    Add it to the sum
    Count it
Divide sum by count to get the average
Print out the average
```

REVIEW: WRITING LOOPS - 3

PROBLEM: Find the average of a set of positive integers entered by the user.

SOLUTION: Pseudocode.

```
Let sum = 0      // The sum of the integers entered by the user.
Let count = 0    // The number of integers entered by the user.
while there are more integers to process:
    Read an integer
    Add it to the sum
    Count it
Divide sum by count to get the average
Print out the average
```

We tell the user to type in zero after all the data have been entered.

What is the problem?

REVIEW: WRITING LOOPS - 4

PROBLEM: Find the average of a set of positive integers entered by the user. The first time the test is evaluated, before the body of the loop has ever been executed, no integer has yet been read!

SOLUTION: Pseudocode. Use a sentinel value - see zyBooks 6.3

Prime the loop

```
Let sum = 0
Let count = 0
Read an integer
while the integer is not zero:
    Add the integer to the sum
    Count it
    Read an integer
Divide sum by count to get the average
Print out the average
```

0 is Sentinel value

Remaining problem: Need test for integer vs character input.

DEMO: `ComputeAverage.java`

Note: If the user enters zero as the first input value, there are no data to process. We can test for this case by checking whether count is still equal to zero after the while loop.

A minor point, but a careful programmer should cover all the bases.

CLICKER QUESTION #1

```
int x = 10;
while( x < 15 ) {
    System.out.print(x + " " );
    x += 2;
}
```

A. 10
B. 3
C. 5
D. Prints 10 endlessly
E. 15

How many times does this loop execute?

CLICKER QUESTION #1 ANSWER

```
int x = 10;
while( x < 15 ) {
    System.out.print(x + " " );
    x += 2;
}
```

How many times does this loop execute?

A. 10
B. 3 correct
C. 5
D. Prints 10 endlessly
E. 15

x is 10
x is 12
x is 14

CLICKER QUESTION #2

```
for(int i = 0; i < 30; i += 3) {
    System.out.println(i + " " );
}
```

How many times does this loop execute?

A. 10
B. 30
C. 5
D. Prints endlessly
E. 29

CLICKER QUESTION #2 ANSWER

```
for(int i = 0; i < 30; i += 3) {
    System.out.println(i + " " );
}
```

How many times does this loop execute?

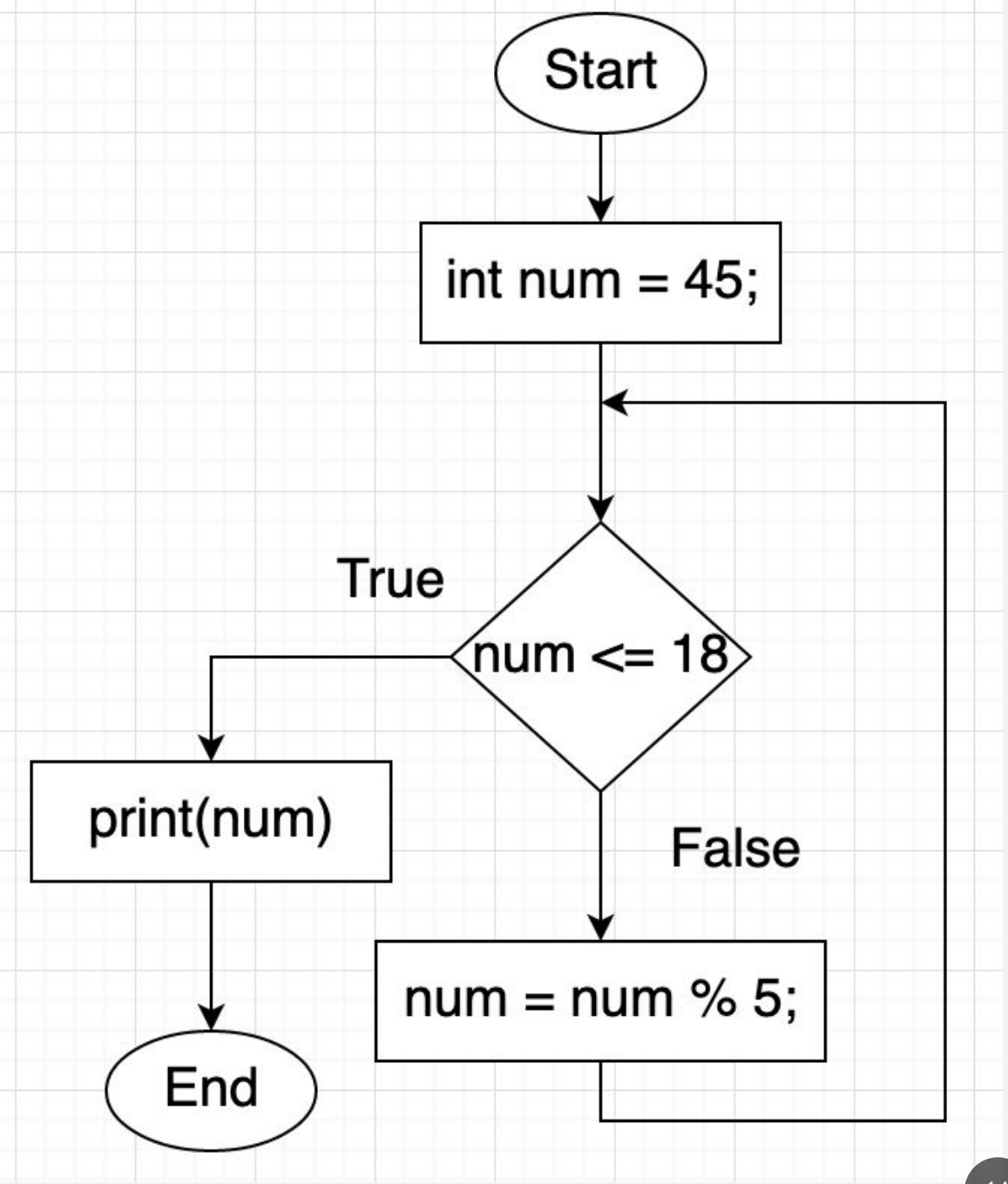
```
0
3
6
9
12
15
18
21
24
27
```

A. 10 correct
B. 30
C. 5
D. Prints endlessly
E. 29

CLICKER QUESTION #3

How many times will the test be executed?

A. 2
B. 8
C. 1
D. 27
E. 0

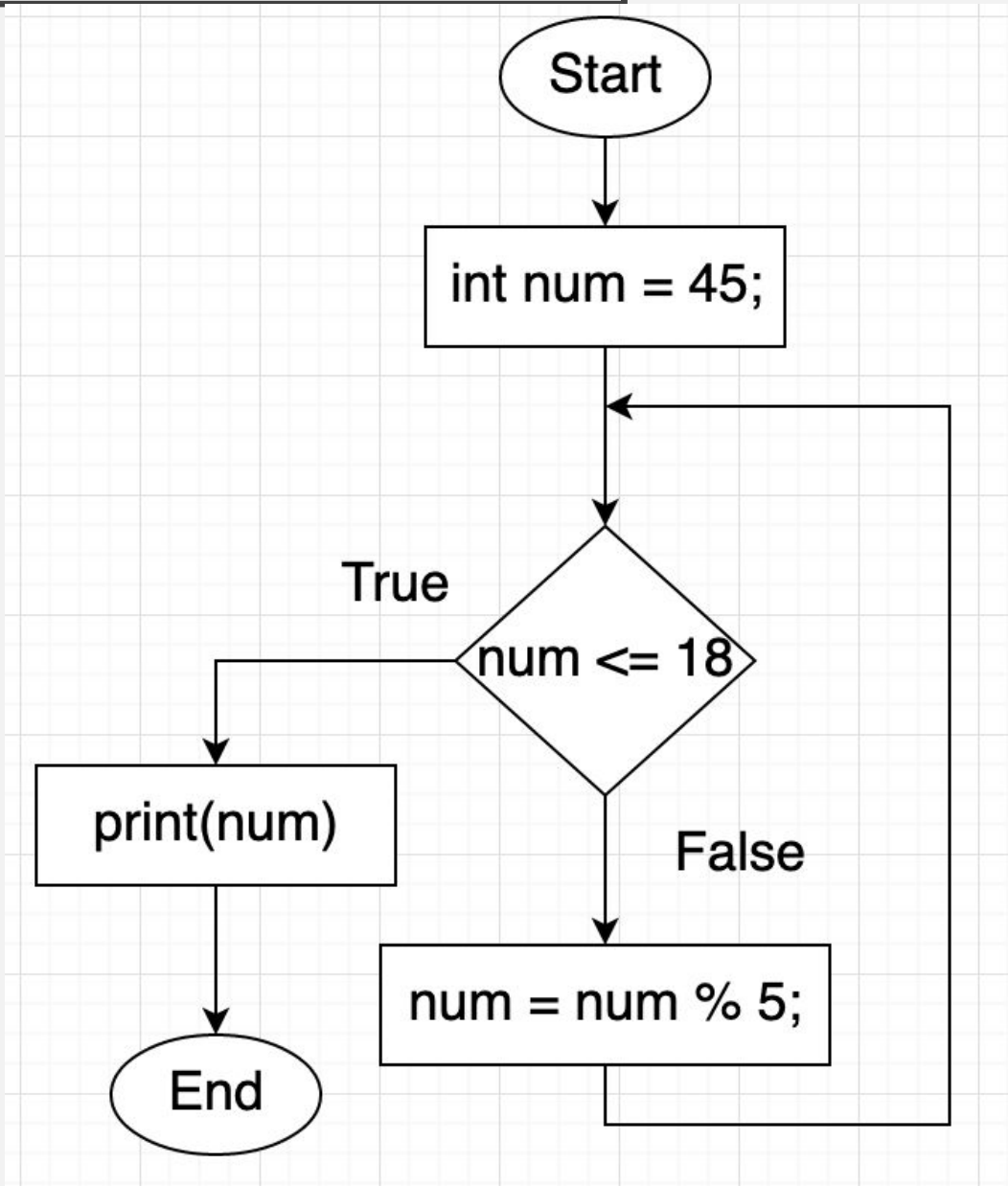


CLICKER QUESTION #3 ANSWER

How many times will the test be executed?

45 <= 18 false
remainder 0 <= 18 true

A. 2 correct
B. 8
C. 1
D. 27
E. 0



STRINGS: REVIEW

- A String is made up of a sequence of characters.
- Each character has an index number- it's position in the sequence.
- Example:

W	o	n	d	e	r	f	u	l	.
0	1	2	3	4	5	6	7	8	9

1. The index starts at 0, *not* 1.
2. The String class charAt method returns a char given an index number.

`char`	**`charAt`**`(int index)` Returns the `char` value at the specified index.

```
String str = "Wonderful.";
char c1 = str.charAt(0);
char c2 = str.charAt(5);
```

c1 is `W`
c2 is `r`

STRINGS AND LOOPS

Problem: Given a String, print the first 5 characters, each on a separate line.
Strategy: use the `charAt` method to access the first five characters in the string and print them using `println`.

Start index = 0, end index = 4.

```
String str = "Wonderful";
for (int i=0; i < 5; i++)
   System.out.println(str.charAt(i));
W
o
n
d
e
```

Problem: Given a String and a character, return true if the character exists in the string, false otherwise.

Input: String and a character.

Strategy: Examine each character in the String and compare it to the target character. If a match is found, return true, if no match found, return false .

Output: true, false

WRITING THE PSEUDOCODE

Write a solution that is closer to Java:

Input: `String inputStr, char target`

Strategy:

```
boolean found = false
for each character c in inputStr
  if c == target
     result = true
return result
```

Writing a solution (algorithm) in code-like form is called "pseudocode".

You can work on the logic and not worry about compiling.

Then, coding in Java is a small step.

JAVA IMPLEMENTATION OF PSEUDOCODE

```java
public static boolean containsChar(String inputStr, char target) {
    boolean found = false;
    char curChar;
    for (int i=0; i<inputStr.length(); i++) {
        curChar = inputStr.charAt(i);
        if (curChar == target)
            found = true;
    }
    return found;
}
```

Think of how to test this code:

1. "Wonderful", 'f' should return true
2. "Wonderful", 'z' should return false

Important points:

1. The `for` loop will access every character in the `inputStr` (note the index start and end numbers).
2. Use of `charAt` and the loop counter to access a char.
3. Use of `==` to test equality with characters.
4. Use of boolean variable to store the result.

DEMO:

`StringLoopDemo.java`

(string methods with for loops and if-else statements)

`IncomeTax.java`

(nested while loops and if-else statements)

CLICKER QUESTION #4

```
public static boolean containsChar(String inputStr, char target) {
    boolean found = false;
    char curChar;
    for (int i=0; i<inputStr.length(); i++) {
        curChar = inputStr.charAt(i);
        if (curChar == target)
            found = true;
    }
    return found;
}
```

On this call:

`containsChar("That's a tasty snack!", 'a');`

How many times will the loop execute?

A. 21
B. 3
C. 2
D. Error
E. 16

CLICKER QUESTION #4

```
public static boolean containsChar(String inputStr, char target) {
    boolean found = false;
    char curChar;
    for (int i=0; i<inputStr.length(); i++) {
        curChar = inputStr.charAt(i);
        if (curChar == target)
            found = true;
    }
    return found;
}
```

The loop examines all 21 characters in `inputStr`.

A. 21 correct
B. 3
C. 2
D. Error
E. 16

On this call:

`containsChar("That's a tasty snack!", 'a');`

How many times will the loop execute?

BREAK STATEMENT

```
public static boolean containsCharEarlyExit(String inputStr, char target) {
    boolean found = false;
    char curChar;
    for (int i=0; i<inputStr.length(); i++) {
        curChar = inputStr.charAt(i);
        if (curChar == target) {
            found = true;
            break;
        }
    }
    return found;
}
```

T-P-S

How would this code differ from the previous method in the number of times the loop executes?

CONTINUE STATEMENT

A `continue` *statement* in a loop causes an immediate jump to the loop condition check.

```
Scanner in = new Scanner(System.in);
int x = -1;
int sum = 0;
while (x != 0) {
    x = in.nextInt();
    if (x <= 0) {
        continue;
    }
    System.out.println("Adding " + x);
    sum += x;
}
```

`continue` statement causes the program to skip over any negative values.

CLICKER QUESTION 5

Write a *for* loop that prints out the even numbers between 2 and 20.

```
1. for (int n = 1; n <= 10; n++) {
       System.out.println( 2*n );
   }

2. for (int n = 2; n <= 20; n = n + 2) {
       System.out.println( n );
   }

3. for (int n = 2; n <= 20; n++) {
       if ( n % 2 == 0 )
       System.out.println( n );
   }

4. for (int n = 1; n <= 1; n++) {
       System.out.println("2 4 6 8 10 12 14 16 18 20");
   }
```

Which of the loops print out the correct answer?

A. 1, 2. 3. 4
B. 1, 2, 3
C. 1, 2, 4
D. 2, 3, 4

CLICKER QUESTION 5 ANSWER

Write a for loop that prints out just the even numbers between 2 and 20.

```
1. for (int n = 1; n <= 10; n++) {
       System.out.println( 2*n );
   }

2. for (int n = 2; n <= 20; n = n + 2) {
       System.out.println( n );
   }

3. for (int n = 2; n <= 20; n++) {
       if ( n % 2 == 0 )
       System.out.println( n );
   }

4. for (int n = 1; n <= 1; n++) {
       System.out.println("2 4 6 8 10 12 14 16 18 20");
   }
```

4. Marked wrong in exam! Do not "hard code"

Which of the loops print out the correct answer?

A. 1, 2. 3. 4
B. 1, 2, 3
C. 1, 2, 4
D. 2, 3, 4

DEMO - NESTED LOOPS

Example: `ListLetters.java`

Write a program that reads a line of text entered by the user. It prints a list of the letters that occur in the text, and it reports how many different letters were found.

```
Please type in a line of text.
The quick brown fox jumps over the lazy dog
Your input contains the following letters:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

There were 26 different letters.
```

Uses nested loops. A ***nested loop*** is a loop that appears in the body of another loop. The nested loops are commonly referred to as the ***inner loop*** and ***outer loop.***

NESTED LOOPS

Outer loop

Inner loop

```
for(letter = 'A'; letter <= 'Z'; letter++) {
    int i; //Position of a character in str.

    for(i = 0; i < str.length(); i++) {
        if(letter == str.charAt(i)) {
            System.out.print(letter + " ");
            count++;
            break;
        }
    }
}
```

OUTER LOOP

Start with letter A and ends with letter Z, incrementing each time by one letter.

INNER LOOP:

Loops through the string after getting its length.

Checks if each character is a letter.

Prints out the letter.

Counts it.

Breaks out of loop after character is counted.

TO-DO LIST:

- Complete zyBook chapter 6 exercises.
- Communicate with us using only Moodle forum or Piazza.
- Complete Project 3 *before* the exam.
- Do all the review worksheets and practice exams.