Sentinel- controlled loops in Java

- The loop and a half: We have a repeating task of some kind, but some parts of the task need to be done one more time than other parts.

Example) Printing a sequence of numbers separated by commas. We have 5 numbers to print but only 4 commas.

One solution to this problem above:

Int num = 1;

While (num < 5)

{

System.out.print(num);

System.out.print(“, “);

num++;

}

System.out.println(num);

The last time a comma is included is when the number is less than 5, or when num is assigned to 4. At the end of the while loop, there is a post decrement that adds 1 to 4 and assigns the value (5) to num. Then, we skip the while loop and print num, which is 5.

- Another version of the loop and a half problem is sentinel-controlled input. The idea here is that we have the user entering values. We don’t know upfront how many values the user will want to enter. So, this could be a list of numbers and then there’s a special value that executes termination of the program. We will not want to process the last value. It is the option that quits. The processing will continue until the quit option is entered.

Basic algorithm for a sentinel-controlled loop:

Ask for value

Read value form keyboard

WHILE that value is not equal to the stopping value

Process the value

Ask for the next value

Read value from keyboard

}

Report results

The "loop and a half" concept means:

1. We need more than one loop
2. A portion of the loop's body needs to be done an extra time, so we also do that part either before or after the loop.
3. The loop will always need to execute more than once.

Suppose we modify the program so that asking for and reading the value from the user happens as the first thing in the loop (not before and not at the end of the loop body) without making any other changes. How will that change the behavior of the program?

A. The sum will be one less than it should be.

B. The sum will be one more than it should be.

C. The sum will still be correct.

// When we enter -1, it will subtract 1 from the sum before it exits the loop.