Intermediate Loops

1. The Legendary FizzBuzz

Write a program that prints the numbers from 1 to 100.

But for multiples of three print "Fizz" instead of the number.

For the multiples of five print "Buzz".

For numbers which are multiples of both three and five print "FizzBuzz".

2. The Halving Machine

Create a loop that will take an integer from user input.

If the integer is odd, add 1 and then divide that by 2 and print the new number.

If the integer is even, just divide by 2 and print the new number.

Continue doing this until you reach 1.

If the user enters 0 or negative number, print out "HEY! That's against the rules." and end the program.

3. Letter at a time

Did you know that using a loop, you can examine a String one letter at a time?

The two key built-in String methods are length() and charAt().

length() returns an int representing the total number of characters in the String

charAt( int n ) returns character at position n of the string (the character positions start at 0)

Try using to print the character and position number for every character in a string from user input!

Your output could look like the following:

What is your message?

>Hello World

Here are all the characters, one at a time:

0: 'H'

1: 'e'

2: 'l'

3: 'l'

4: 'o'

5: ' '

6: 'W'

7: 'o'

8: 'r'

9: 'l'

10: 'd'

4. Coordinates in a plane

Create a program that takes two integers: [x] and [y].

In that program, print out all possible coordinates of (0-x, 0-y)

Note that 0-y means 0 through y in this case.

Your output could look like the following:

What is the maximum x?

>2

What is the maximum y?

>3

Here are your possible coordinates:

(0,0)

(0,1)

(0,2)

(0,3)

(1,0)

...

(2,3)

5. Create a JAVA program to write the even numbers from 10 to 20, both included, except 16, in 3 different ways:

- Incrementing 2 in each step (use "continue" to skip 16)

- Incrementing 1 in each step (use "continue")

- With and endless loop (using "break" & "continue")

6. Write a program that will evaluate simple expressions such as 17 + 3 and 3.14159 \* 4.7. The expressions are to be typed in by the user. The input always consist of a number, followed by an operator, followed by another number. The operators that are allowed are +, -, \*, and /. Your program should read an expression, print its value, read another expression, print its value, and so on. The program should end when the user enters 0 as the first number on the line.

7. Create a program using nested loops that would generate the following output:

1.

88888888

88888888

88888888

88888888

88888888

2.

8888

888

88

8

3.

8

8

8

8

8

8