

CSCI 1301 – Lab 16

September 30, 2019

1 Increment and Decrement Operators

Execute (a variation of) the code we just reviewed to make sure you understand the mechanism of the increment and decrement operators.

```
1      int a = 0, b = 0;
2      Console.WriteLine("Before changing their values:");
3      Console.WriteLine($"\\ta is {a}\\n\\tb is {b}\\n-----");
4      Console.WriteLine("Incrementing, using postfix and prefix operators:");
5      a++;
6      ++b;
7      Console.WriteLine($"\\ta is {a}\\n\\tb is {b}\\n-----");
8      Console.WriteLine("Decrementing, using postfix and prefix operators:");
9      a--;
10     --b;
11     Console.WriteLine($"\\ta is {a}\\n\\tb is {b}\\n-----");
12     Console.WriteLine("When combining decrementing and incrementing operators
↪ with other operations, \\nit makes a difference to use postfix or prefix
↪ operators!");
13     int c = a--, d = ++b;
14     Console.WriteLine($"\\ta is {a} (the decrementing took place as
↪ expected)\\n\\tb is {b} (the incrementing took place as expected)\\n\\tc is
↪ {c}" +
15     $" (c got its value *before* a was decremented)\\n\\td is {d} (d got its
↪ value *after* b was incremented)\\n-----");
```

2 First While Loops

1. Write a while loop that displays the integers between 1 and 100 at the screen, with a space between them.
2. Write a while loop that displays the "*" character 100 times at the screen.
3. Modify your previous loop, so that a new line character is displayed on the screen every time 10 "*" has been displayed on the screen. That is, your program should display on the screen:

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

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

3 User Input Validation

3.1 Integer Validation

Consider the following code:

```
Console.WriteLine("Please enter a positive number");
int n = int.Parse(Console.ReadLine());
while (n < 0)
{
    Console.WriteLine($"You entered {n}, I asked you for a positive number.
    Please try again.");
    n = int.Parse(Console.ReadLine());
}
```

1. As always, start by creating a blank project, copy-and-paste that “snippet” into the Main method, compile it and execute it.
2. Then, copy and, comment it out, and adapt your copy so that the user will be asked to enter an integer between 0 and 100, and asked again as long as (s)he does not comply.
3. Re-do the previous step, but change the condition, so that the user has to enter an even number.

3.2 String Validation

Adapt the code above to perform string validation: ask the user to enter a string, and as long as the user does not enter “Yes” or “No”, ask him/her again to enter a value.

4 Conditional Operator (optionnal)

There is an operator for `if else` statements for particular cases (assignment, call, increment, decrement, and new object expressions). You can read about it at <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/conditional-operator>, its structure is:

```
condition ? first_expression : second_expression;
```

An example could be:

```
1 bool adult;
2 // Assume that adult is set to true or false.
3 int price = adult ? 5 : 3;
```

If `adult` is true, then `price` will get the value 5, otherwise, `price` will get the value 3.

Go back to lab 14¹ and try to rewrite some of the portions of code you wrote using this operator.

¹../14/