

CSCI 1301 – Lab 11

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 Flavors of while loops

- Write a while loop asking the user for an integer, and looping as long as that integer is strictly greater than 0.
- Transform your while loop so that the program displays the sum of all the numbers entered when exiting the while loop.

An example of execution could be:

```
Enter a positive number to sum, or a negative number to exit
12
Enter a positive number to sum, or a negative number to exit
3
Enter a positive number to sum, or a negative number to exit
-2
Your total is 15.
```

Identify what type of loop it is: is it user-controlled, counter-controlled, or sentinel-controlled? Does it use a counter, a sentinel value, or an accumulator?

5 TryParse Method

5.1 Getting Familiar With It

Consider the code we just studied:

```
Console.WriteLine("Please, enter an integer.");
string message = Console.ReadLine();
int a;
bool res = int.TryParse(message, out a);
if (res)
{
    Console.WriteLine($"The value entered was an integer: {a}.");
}
else
{
    Console.WriteLine("The value entered was not an integer, so 0 is
↪ assigned to a.");
}
Console.WriteLine(a);
```

What happen if:

- The user enters an integer?
- The user enters a floating-point value?
- The user enters nothing?
- The user enters a string that contains alphabetical characters?

5.2 Using It

1. Write a code that ask the user to enter an integer, and ask the user again as long as the user entered something that isn't an integer.
2. Actually, there is a TryParse method in other classes as well: there is for instance a Double.TryParse and a Decimal.TryParse method. Write a small program that uses one of them.