

CSCI 1301 - Lab 04

Clément Aubert

August 28, 2018

Deadlines: This lab needs to be completed before taking Lab 05.

Dependencies: This lab can be done in almost any order, but the recommended order is of course the one below.

Part 0 - Preamble

Making Sure You Saved the Right Files

The title of this part sums it all: re-downlad the files you saved for lab 03 and open the project with Visual Studio. Try to compile and execute it. Now, create a folder for this lab: upon completion of this lab, you will have created (at least) three new projects, that should each have one sub-folder. Make sure you save all of them correctly before exiting the lab.

Reading the Instructions

From now on, read “Create a new project” as

Create a new project, using the “Console Application Visual C#” template, as you did in a. and b. of Part I of Lab 03. Pick simple and “valid” names for your project and solution, and make sure you save it in the right place. Compile and execute your program frequently, for instance upon completion of every item. *Do not hesitate to change your program to answer questions: you’re in a lab, you’re supposed to conduct experiments!* If you get an error when trying to compile your program, use Ctrl + Shift + F12 to “jump” to the line where VS thinks there is an error.

Part I - More Printing and Escaping

- a. Create a new project.
- b. Edit the Main method, so that when compiled and executed, you program will print the following:

```

!
!!!
!!!!!!!

```

Press any key to continue . . .

- c. We will now use the “Find and Replace” feature of VS (cf. https://msdn.microsoft.com/en-us/library/139eef4h.aspx#Anchor_1). Hit “Edit” → “Find and Replace” → “Find in Files”, or press Ctrl + Shift + f. In the panel that appears, click on the “Replace in Files” tab, enter “!” (without the quotes) under “Find what:”, and “***” (without the quotes) under “Replace with:”. Hit “Replace All”, and note the modification in your program.
- d. As you can see this is a really useful feature of VS, but also a really dangerous one. If you were to replace all the “***” characters with “!” in all the program we wrote so far, what could possibly go wrong?
- e. Edit one last time your program to print the following:

```

                *
            *   *   *
*   *   *   *   *   *   *

```

Press any key to continue . . .

Can you think of a way to change your program using the “Find and Replace” feature of VS to do that?

Part II - First Variables Manipulation

- a. Create a new project
- b. In the Main method, add three statements:
 - One that declares a string variable named `lastName`,
 - One that declares a string variable named `firstName`,
 - One that declares an int variable named `classOf`
- c. Assign your last name to the first variable, your first name to the second variable, and your anticipated graduation year (i.e., 2020, for instance) to the third one, using three statements.
- d. Print the values of the three variables, using the following statement (that uses interpolation):

```

Console.WriteLine($"My name is {firstName} {lastName}, and I expect to graduate in
↪ {classOf}.");

```

- e. Answer the following: What would happen if the statement of d. was *before* the assignments of c., i.e., after the declarations of the variables, but before their assignments?
- f. After the printing statement of d., add three statements that change the value of the three variables, and copy the statement of d. Notice that the very same statement will now print a different message!

Part III - Variables Assignments

- a. Create a new project
- b. In the Main method, add the following:

```
int a = 12;  
int b;  
b = a;  
a = 24;
```

- c. Now, answer the following: What are the values of a and b...
 - ... after the statement on line 2 is executed?
 - ... after the statement on line 3 is executed?
 - ... after the statement on line 4 is executed?

Use printing instructions to check your answer, and make sure you understand how variable assignment is done.

Part IV (Optional) - Pushing Further

The following are two independent tasks, to widen your understanding of this class, and to prepare you for the next labs.

- a. If you read carefully <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/interpolated-strings>, you may notice that there is another way to print variables. Indeed, the code of Part II could be replaced with

```
Console.WriteLine("My name is {0} {1}, and I expect to graduate in {2}.", firstName,  
    ↪ lastName, classOf);
```

This is called *composite formatting*. Try to do the following:

- Create a project where you use this formatting.
- Print the { character in a Console.WriteLine statement that does not use composite formatting.
- Print the { character in a Console.WriteLine statement that uses composite formatting.

You should have a look at the documentation at <https://docs.microsoft.com/en-us/dotnet/standard/base-types/composite-formatting#escaping-braces> if you can't figure out how to do it.

- b. Pick one of your program, and comment out the using System; statement. If you're using Console.WriteLine or Console.Write instructions, you should not be able to compile your program anymore. Try replacing these instructions (using "Find and Replace"!) with System.Console.WriteLine and System.Console.Write. What do you observe? Can you explain this behavior?