

CSCI 1301 – Lab 04

August 26, 2019

1 Reading the Instructions

This is the fourth lab already, and you should have some habits. You should know how to

1. Manipulate projects (creating them, opening them, editing them, saving them, making backups).
2. Compile and execute your program frequently.

You already started to read “compile and execute” as

In the menu, click on `Build` → `Build solution` (or use `Ctrl + Shift + B`), then on `Debug` → `Start without Debugging` (or use `Ctrl + F5`).

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

Create a new project, using the “Console Application Visual C#” template, as you did in lab 03¹. 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.

And, *even if it is not explicitly asked*, you should save your work once you are done, and re-open it at the beginning of the next session to make sure you saved it properly.

If you want a tutor or your instructor to check your answer to a previous lab, just ask!

2 Exploring the Documentation

The documentation for Visual Studio and C# is packed with useful information, and efforts are made to make it accessible to beginners. The goal of this exercise is to help you realizing that it contains answers to questions you may have asked yourself, like “what is a solution?”, or “what does the namespace keyword do?”.

All the documentation for Visual Studio is at <https://docs.microsoft.com/en-us/visualstudio/>. The documentation for C# is at <https://docs.microsoft.com/en-us/dotnet/csharp/>. To get started, have a look at the first three paragraphs of <https://docs.microsoft.com/en-us/visualstudio/ide/creating-solutions-and-projects> (or at <https://docs.microsoft.com/en-us/visualstudio/mac/projects-and-solutions?view=vsmac-2019> for Mac), and answer the following:

1. What is a solution?
2. What is a project?
3. Which one contain the other: the solution, or the project?

¹[../03/#starting-from-a-template](#)

Note that since we will only manipulate one project at a time, the distinction between solution and project will not be critical in this lab. That's the reason why I wrote previously "create a new project" with the sense of "create a new solution and a new project in it": we will never manipulate more than one project in a solution at a time.

Finally, read the page at <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/namespaces/>: do you know an example of namespace that we used?

3 Variables Assignments

Think with a pen and a sheet of paper for some time before opening Visual Studio. Consider the following code:

```

1  int a = 0;
2  a = 12 - 3;
3  a = 10 % 3;
4  a = 12 * 2;
5  a = 9 / 3;
6  a = -3;
7  a -= 3;
8  a += 5;
9  a *=12;
10 a /= 2;
11 a = a + a;
```

1. In the previous code, what is the value of a after each line is executed?
2. Check your answers: create a new project, copy the previous statements in the Main method, use `Console.WriteLine(a);` to display the value of a after each statement.

4 More About Displaying Characters at the Screen

1. Create a new project.
2. Edit the Main method, so that when compiled and executed, your program will display the following at the screen:

```

1      !
2      !!!
3      !!!!!!!
4
5  Press any key to continue . . .
```

1. We will now use the "Find and Replace" feature of VS².
 - a) Hit "Edit" → "Find and Replace" → "Find in Files", or press Ctrl + Shift + f.
 - b) 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:".
 - c) Hit "Replace All", and note the modification in your program.

²https://msdn.microsoft.com/en-us/library/139eef4h.aspx#Anchor_1

2. 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?