

CSCI 1301 - Lab 07

Clément Aubert

January 29, 2018

Deadlines: This lab needs to be completed and perfectly understood before exam #1 (02/15).

Dependencies: This lab uses the knowledge you acquired in all the previous lab.

Part I - Milestone #1

You accomplished a lot since you started this class. Let's take a moment to look back at what you learned, and to make sure you understand all those notions and skills. If you have a doubt, feel free to look back at the corresponding lab. If you haven't already, look at the "Pushing Further" parts of those previous lab, as noted below.

What You Learned

To Use Visual Studio

- What is the structure of a project (Lab 2 - Part I)
- How to open a pre-existing project (Lab 2 - Part III)
- How to open a template and edit it (Lab 3 - Part I)
- How to re-name a project or its files (Lab 2 - Part V)
- To avoid modifying the files of a project outside of VS (Lab 2 - Part V)
- How to compile and execute your program, using a debugger (Lab 2 - Part III and Lab 6 - Part I)
- How to zip and share a project (Homework #1)
- How to use the "Find and Replace" feature of VS (Lab 4 - Part I)

To Be Methodical

- Not to be afraid of the documentation (Lab 2 - Part II)
- To use the following shortcuts (remember that there is a cheatsheet at <http://visualstudioshortcuts.com/>):
 - ALT + F4 to close VS (or any Windows program, actually)
 - Ctrl + Shift + b to build the solution

- Ctrl + F5 to start the program without debugging
- F10 or F11 to start the program with the debugger
- Ctrl + Shift + n to create a new project
- Ctrl + Shift + F12 to “jump” to the line where VS thinks there is an error
- Ctrl + Shift + f to open the “Find and Replace” menu
- CTRL + k, then CTRL + d to indent your code
- To compile your code frequently (Lab 4 - Part 0)
- To read the error messages (Lab 2 - Part IV)
- To organize your files (Lab 3 - Part 0)

To Write Programs!

- What are the important parts of a program (Lab 2 - Part IV)
- To print messages, using `Write` and `WriteLine` (Lab 3 - Part I)
- To print messages that includes escape sequences (Lab 3 - Part I and Lab 4 - Part I)
- To declare, assign and print variables (Lab 4 - Part II and Lab 4 - Part III)
- To read a string from the user (Lab 5 - Part I)
- To convert a string into integer (Lab 5 - Part II)
- To manipulate various numeric datatypes (Lab 6 - Part II)
 - To choose the appropriate numeric datatype
 - To determine the “legality” of an operation involving two different datatypes
 - To use the cast operator to explicitly convert between datatypes
 - To be able to identify the datatype of literals
- To read numeric values from the user (Lab 6 - Part III)

Academic Life

- Not to be afraid of your professor (hopefully!).
- How, where, and when to ask for help.
- What are my expectations for this class.
- How to organize your workflow, the importance of planning ahead.
- ... to be continued!

Maybe you decided what your major was going to be. Maybe you changed your mind. Maybe you're not sure. Being confused and uncertain is sometimes part of the process of taking decisions and learning, and that's all right. It is normal to hesitate. This page by a colleague in Computer Science may be a good read for students hesitating between IT, CS and MIS, or wanting to have more information about those majors.

Did You Pushed Further?

If you haven't done the optional parts of the lab, it may be a good moment to go back and look at them. Computer Science and Information Technology (aspiring) majors are strongly recommended to look at those

“Pushing Further” parts. Those parts help you to be ready for the future: they prepare you for the next lab, give you heads-up on the lecture, and tease your curiosity. In them, you learned (or you’ll learn):

- How to classify programming languages (Lab 2)
 - What is the difference between .NET Core and .NET framework (Lab 3)
 - How to print unicode characters (Lab 3)
 - How to use composite formatting (Lab 4)
 - How to use namespaces in a different way (Lab 4 and Lab 5)
 - How to convert an int, a float and a double into a string (Lab 5 and Lab 6)
- a. Open one of your project. It can be any project, as long as it uses variables.
 - b. Make sure you can compile and run it.
 - c. Hit F11 once. Observe what happen. Move the black screen where the printing happen (called the *console*, or *Command prompt*) to see your code. Hit F11 a couple of times, and see what happen. You are executing your code line by line, and can track many useful information, including the value and datatype of your variables (in the “Locals” panel / window).
 - d. You can find out at <https://msdn.microsoft.com/en-us/library/y740d9d3.aspx> (VS 2015) or <https://docs.microsoft.com/en-us/visualstudio/debugger/navigating-through-code-with-the-debugger> (VS 2017) alternative ways of starting the debugging functionalities of VS.

Part II - Working on Problem-Solving

Here is a problem that involves almost all the previously mentioned notions and skills. It is phrased in a more abstract way, closer to the kind of problem you will be facing if you were a software-developer. Think about what you need to do before starting to type your code, and when you start writing your code, make sure you compile it frequently.

Write a program that asks the user for their name, their number of guests, and the number of pizzas they have. Your program should then print the name of the user, the number of people at the party, the number of pizzas, and what portion of pizza will everyone have, assuming the pizzas will be shared equitably.

A couple of additional precisions:

- The user of the program also want some pizza, so the number of people at the party is the number of guests plus one.
- Once you’re done, you need to test your program:
 - What happen if the user provide “normal”, plausible data? Does your program work as expected?
 - What happen if the name of the user is a numeric value?
 - What happen if the user have 2 guests and 4 pizza? Will everyone get $4 / 3 = 1 + \frac{1}{3}$ pizza, as they normally should?
 - What happen if the user have 0 pizzas?
 - What happen if the user have 0 guests?
 - What happen if the user have -4 pizzas?

- What happen if the user have -3 guests?

It would be nice if you could use (some of) the following techniques:

- Combined assignment operator
- Unicode printing (make it stylish!)
- Cast operator (the number of guests is an integer, and you could think that bringing non-whole pizzas at a party could be weird, so that the number of pizza would also need to be an integer)
- String concatenation