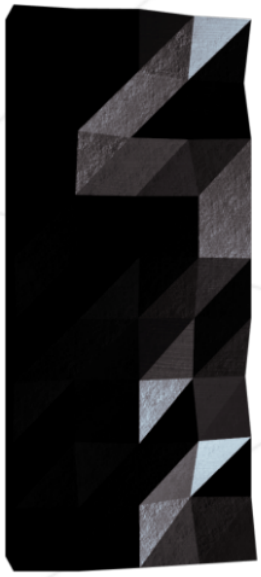

C++ - Module 00

N mesp ce, cl ss, member functions, stdio stre m, initi liz tion lists, st tic, const, nd lots of b sic stuff

Summ ry: This document cont ins the subject for Module 00 of the C++ modules.

Version: 7

Chapter I

General rules

For the C++ modules you will use and learn C++98 only. The goal is for you to learn the basic of an object oriented programming language. We know modern C++ is way different in a lot of aspects so if you want to become a proficient C++, developer you will need modern standard C++ later on. This will be the starting point of your C++ journey it's up to you to go further after the 42 Common Core!

- ny function implemented in a header (except in the case of templates), and any unprotected header means 0 to the exercise.
- Every output goes to the standard output and will be ended by a newline, unless specified otherwise.
- The imposed filenames must be followed to the letter, as well as class names, function names, and method names.
- Remember: You are coding in `C++` now, not in `C` anymore. Therefore:

  The following functions are FORBIDDEN, and their use will be punished by a `0`, no questions asked: `*alloc`, `*printf` and `free`.

  You are allowed to use everything in the standard library. HOWEVER, it would be smart to try and use the C++-ish versions of the functions you are used to in C, instead of just keeping to what you know, this is a new language after all. nd NO, you are not allowed to use the STL until you are supposed to (that is, until module 08). That means no vectors/lists/maps/etc... or anything that requires an include <algorithm> until then.

- ctually, the use of any explicitly forbidden function or mechanic will be punished by a `0`, no questions asked.
- lso note that unless otherwise stated, the `C++` keywords `"using namespace"` and `"friend"` are forbidden. Their use will be punished by a `-42`, no questions asked.
- Files associated with a class will always be `ClassName.hpp` and `ClassName.cpp`, unless specified otherwise.
- Turn-in directories are `ex00/`, `ex01/`, ..., `exn/`.

- You must read the examples thoroughly. They can contain requirements that are not obvious in the exercise's description.
- Since you are allowed to use the `C++` tools you learned about since the beginning, you are not allowed to use any external library. nd before you ask, that also means no `C++11` and derivatives, nor `Boost` or anything else.
- You may be required to turn in an important number of classes. This can seem tedious unless you're able to script your favorite text editor.
- Read each exercise FULLY before starting it! Do it.
- The compiler to use is `c++`.
- Your code has to be compiled with the following flags : `-Wall -Wextra -Werror`.
- Each of your includes must be able to be included independently from others. Includes must contain every other includes they are depending on.
- In case you're wondering, no coding style is enforced during in `C++`. You can use any style you like, no restrictions. But remember that a code your peer-evaluator can't read is a code they can't grade.
- Important stuff now: You will NOT be graded by a program, unless explicitly stated in the subject. Therefore, you are afforded a certain amount of freedom in how you choose to do the exercises. However, be mindful of the constraints of each exercise, and DO NOT be lazy, you would miss a LOT of what they have to offer
- It's not a problem to have some extraneous files in what you turn in, you may choose to separate your code in more files than what's asked of you. Feel free, as long as the result is not graded by a program.
- Even if the subject of an exercise is short, it's worth spending some time on it to be sure you understand what's expected of you, and that you did it in the best possible way.
- By Odin, by Thor! Use your brain!!!

Contents

I	**General rules**	**1**
II	**Exercise 00: Megaphone**	**4**
III	**Exercise 01: My wesome PhoneBook**	**5**
IV	**Exercise 02: The Job Of Your Dreams**	**7**

Chapter II

Exercise 00: Megaphone

	Exercise : 00
Exercise 00: Megaphone	
Turn-in directory : *ex*00	
Files to turn in : `Makefile, megaphone.cpp`	
Forbidden functions : `None`	

Just to be sure that everybody is awake, write a program that has the following behavior:
(Use C++ style to interact with stdin!)

```
$>./megaphone "shhhhh... I think the students are asleep..."
SHHHHH... I THINK THE STUDENTS  RE  SLEEP...
$>./megaphone Damnit " ! " "Sorry students, I thought this thing was off."
D MNIT ! SORRY STUDENTS, I THOUGHT THIS THING W S OFF.
$>./megaphone
* LOUD  ND UNBE R BLE FEEDB CK NOISE *
$>
```

Chapter III

Exercise 01: My wesome PhoneBook

	Exercise : 01
Exercise 01: My wesome PhoneBook	
Turn-in directory : *ex*01	
Files to turn in : `Makefile, *.cpp, *.{h, hpp}`	
Forbidden functions : `None`	

Welcome to the 80s and its unbelievable technology! Write a program that behaves like a `crappy` awesome phonebook software. Please take some time to give your executable a relevant name. When the program starts, the user is prompted for input: you should accept the `DD` command, the `SE RCH` command or the `EXIT` command. ny other input is discarded.

The program starts empty (no contacts), doesn't use any dynamic allocation, and can't store more than 8 contacts. If a ninth contact is added, replace the oldest contact.

http://www.cplusplus.com/reference/string/string/ and of course http://www.cplusplus.com/reference/iomanip/

- If the command is `EXIT`:

  The program quits and the contacts are lost forever.

- Else if the command is `DD`:

  The program will prompt the user to input a new contact's information, one field at a time, until every field is accounted for.

  contact is defined by the following fields: `first name`, `last name`, `nickname`, `phone number`,`darkest secret`.

  The PhoneBook must be represented as as an instance of a class in your code it must contain an array of contact.

  contact MUST be represented as an instance of a class in your code. You're free to design the class as you like, but the peer evaluation will check the consistency of your choices. Go look at today's videos again if you don't understand what I mean (and I don't mean "use everything" before you ask).

- Else if the command is `SE RCH`:

  The program will display a list of the available non-empty contacts in 4 columns: `index`, `first name`, `last name` and `nickname`.

  Each column must be 10 chars wide, right aligned and separated by a `'|'` character. ny output longer than the columns' width is truncated and the last displayable character is replaced by a dot (`'.'`).

  Then the program will prompt again for the index of the desired entry and displays the contact's information, one field per line. If the input makes no sense, define a relevant behavior.

- Else the input is discarded.

When a command has been correctly executed, the program waits for another `DD` or `SE RCH` command until an `EXIT` command.

Chapter IV

Exercise 02: The Job Of Your Dreams

	Exercise : 02
Exercise 02: The Job Of Your Dreams	
Turn-in directory : *ex*02	
Files to turn in : `Makefile, ccount.cpp, ccount.hpp, tests.cpp`	
Forbidden functions : `None`	

It's your first day of work at `GlobalBanksters United`. You successfully passed the hiring tests for the programmers' team thanks to a few tricks with `Microsoft Office` a friend showed you. But you know that it was your swift installation of `dobe Reader` that blew your recruiter's mind. This gave you the little edge needed to beat your opponents for this job.

nyway, you made it and your boss gave you your first task. You need to recode one missing source file because something went wrong. `ccount.cpp` is missing and they use USB file sharing and not git.

t this point, it will be legitimate to quit this place however for the sake of this exercise, you will stay.

The `ccount.hpp` file is present and a quick compilation of `tests.cpp` confirms that an `ccount.cpp` file is missing. There's also an old output log that seems to contain the matching output.

So you need to create a `ccount.cpp` file and quickly write some lines of pure awesome `C++` and after a couple of failed compilations, your program will compiles and passes the tests with a perfect output, except for the timestamps differences. Damn you're good!