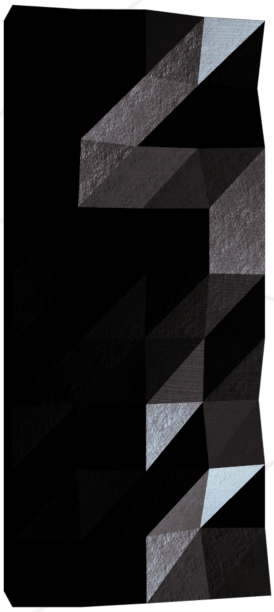

C++ Pool - d00

Namespace, class, member functions, stdio stream, initialization lists, static, const, and lots of basic stuff

Staff 42 bocal@staff.42.fr

Abstract: This document contains the subject for day 00 of 42's C++ pool.

Contents

Chapter I

General rules

- Any 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 `-42`, no questions asked: `*alloc`, `*printf` and `free`.
  - You are allowed to use basically 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. And NO, you are not allowed to use the STL until you actually are supposed to (that is, until d08). That means no vectors/lists/maps/etc... or anything that requires an include <algorithm> until then.
- Actually, the use of any explicitly forbidden function or mechanic will be punished by a `-42`, no questions asked.
- Also 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. If something seems ambiguous, you don't understand `C++` enough.

- Since you are allowed to use the `C++` tools you learned about since the beginning of the pool, you are not allowed to use any external library. And before you ask, that also means no `C++11` and derivates, nor `Boost` or anything your awesomely skilled friend told you `C++` can't exist without.

- 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 ! Really, do it.

- The compiler to use is `clang++`.

- 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 contains every other includes they are depending on, obviously.

- The subject can be modified up to 4h before the final turn-in time.

- In case you're wondering, no coding style is enforced during the `C++` pool. You can use any style you like, no restrictions. But remember that a code your peer-evaluator can't read is a code she or he can't grade.

- Important stuff now : You will NOT be graded by a program, unless explictly 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 day 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 absolutely 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 !!!

Chapter II

Foreword

Here we are, it's finally time for the `C++` pool. As I'm pretty sure your tastes in music are still insanely bad, here are a few good bands you should listen to. If you can't find anything you like in this list, you should seriously consider not listening to any music at all. Ever.

- The Ocean
- Ayreon part 1 and part 2
- Ulver
- Glaciation part1 and part2
- Drudkh
- Tides Of Man
- Tool

In case you missed them the first time, here are the bands I recommended in the `libft` project. Better late than never.

- Between The Buried And Me
- Tesseract
- Chimp Spanner
- Emancipator
- Cynic
- Kalisia
- Wintersun
- O.S.I
- Dream Theater
- Pain Of Salvation
- Crucified Barbara

Chapter III

Exercise 00: Megaphone

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

Just to be sure that everybody is awake, write a program that has the following behavior :

```
$>./megaphone "shhhhh... I think the students are asleep..."
SHHHHH... I THINK THE STUDENTS ARE ASLEEP...
$>./megaphone Damnit " ! " "Sorry students, I thought this thing was off."
DAMNIT ! SORRY STUDENTS, I THOUGHT THIS THING WAS OFF.
$>./megaphone
* LOUD AND UNBEARABLE FEEDBACK NOISE *
$>
```

Bonus: bring an actual megaphone and report to Zaz.

Chapter IV

Exercise 01: My Awesome PhoneBook

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

Welcome in 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 `ADD` command, the `SEARCH` command or the `EXIT` command. Any 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, please define a relevant behavior.

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 `ADD`:
  - The program will prompt the user to input a new contact's informations, one field at a time, until every field is accounted for.
  - A contact is defined by the following fields : `first name`, `last name`, `nickname`, `login`, `postal address`, `email address`, `phone number`, `birthday date`, `favorite meal`, `underwear color` and `darkest secret`.
  - A 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 `SEARCH`:
  - 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. Any 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 `ADD` or `SEARCH` command until an `EXIT` command.

Once done, ask students around you to test your `ADD` command. That way you'll know their darkest secret by using the `SEARCH` command after they left, which is obviously the only relevant part of this exercice. They will have no clue of your evil plot, huehehe.

Chapter V

Exercise 02: The Job Of Your Dreams

	Exercise 02
Exercise 02: The Job Of Your Dreams	
Turn-in directory : $ex02/$	
Files to turn in : `Account.class.cpp`	
Forbidden functions : `None`	
Remarks : `n/a`	

It's your first day of work at `GlobalBanksters United`. You succesfully 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 `Adobe Reader` that really blew your recruiter's mind. This gave you the little edge needed to beat you opponents for this job.

Anyway, you made it and your boss gave you your first task. He explained you that somebody lost the source code that handles the accounts of the bank's customers. Your boss decided to recompile and restart the program to get the file back. Your predecessor got fired because he tried to explain to your boss how do to his job: he mentionned that "this is not how it's supposed to work", "you deleted that file asshole !" and "we should have used Git as I told you". What a snotty little prick, wasn't he ?

After a forty minutes long rant on your predecessor's lack of expertise and constant bullshiting, you are assigned with the writing of the missing source file for tomorrow. Your boss would have loved to do it by himself, but you know, he has managerial things to do. So, he sent you the `Account.class.hpp` file attached to an email protected with a `gpg` key, along with his private key. "Security is important, the hackers can strike anywhere" said your boss as a conclusion as you leave his office.

Uncomfortable in your suit and sweating profusely, you walk by the printer and the photocopier in the long hallway towards the open space to find your desk. You're in the center area and everybody can see your screen. You also notice the big sign on the wall that happilly and colorfully states "Headphones prevent team building ! Silence enhances team building !". It's going to be a long day.

As you sit at your desk, you notice the `Windows XP` login window on your screen. Nobody gave you any login/pass this morning, so you ask very politely to your neighbor if he knows where to get them. His answer, "For questions, write a ticket, everybody knows that, DUH !", doesn't really help you. Plus, you suspect that tomorrow you'll discover that you're also in charge of the tickets because you seem to be the only IT employee in the building. Without anything to lose, you try `admin/admin` on the login window. The computer unlocks and after a minute or so, the icons appear and you can use the computer.

Then, you spend a couple hours figuring things out. You understand that `PuTTy` allows you to `ssh` as root without any password on a server somewhere. The server is an old Ubuntu server and seems to run most of the company's services. You're able to create an account for you with the relevant credentials in order to stop to log as root and you block ssh as root. You also notice that you don't have any mail account and as a consequence, you'll never receive the email with the attached file from your boss.

Thanks to your `bash-fu`, you are finally able to locate the sources of the project that your are supposed to fix in a deep folder named `"test2_REAL_ONE_DONT_DELETE/"`. The folder contains a few files written between 1989 and 1992 by a guy named Brad MacLane. The `Account.class.hpp` file is present and a quick compilation confirms that an `Account.class.cpp` file is missing. There's also a file with some tests, and an old output log that seems to contain the matching output.

Then you quickly write about 150 lines of pure awesome `C++` and after a couple failed compilations, your program compiles and passes the tests with a perfect output, except for the timestamps. Damn you're good ! But your troubles are not over as you hear your boss shouting from the hallway: "who the fuck messed with the production server ! I can't log in anymore !"

Bonus : Add an attribute to the class that counts the number of times the member function "int checkAmount(void) const;" is called. Do that without changing anything to the prototype of this member function. You will need a new keyword, and this exemple is the perfect situation to introduce this new keyword... Of course this keyword is not in the videos.