Go to homepage 🏠 (../../)

SCALE FOR PROJECT CPP MODULE 03 (HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/CPP-MODULE-03)

You should evaluate 1 student in this team

Introduction

Please respect the following rules:

- Stay polite, courteous, respectful, and constructive during the evaluation process. The well-being of the community depends on it.

- Identify with the evaluated person or group any malfunctions in their work. Take the time to discuss and debate the identified problems.

- Keep in mind that there may be slight differences in interpretation between the project instructions, its scope, and functionalities. Keep an open mind and rate as honestly as possible. Pedagogy is only valid if peer evaluation is taken seriously.

Guidelines

- Only rate what is contained in the cloned Git repository of the student or group.

- Verify that the Git repository belongs to the student or group, that the project is the expected one, and that "git clone" is used in an empty folder.

- Thoroughly check that no alias has been used to deceive you and ensure that you are evaluating the official submission.

- To avoid any surprises, check with the student or group any potential scripts used to facilitate evaluation (e.g., test or automation scripts).

- If you have not done the project you are evaluating, you must read the subject entirely before starting the evaluation.

- Use the available flags to report an empty submission, a non-functioning program, a Norm error, cheating, etc.

Attachments

subject.pdf (https://github.com/rphlr/42-Subjects/) Account.hpp (https://github.com/rphlr/42-Subjects/) 19920104_091532.log (https://github.com/rphlr/42-Subjects/) tests.cpp (https://github.com/rphlr/42-Subjects/)

Go to homepage 🏠 (../../)

Preliminary tests

If cheating is suspected, the evaluation stops here. Use the "Cheat" flag to report it. Take this decision calmly, wisely, and please, use this button with caution.

Prerequisites

The code must compile with c++ and the flags -Wall -Wextra -Werror
Don't forget this project has to follow the C++98 standard. Thus,
C++11 (and later) functions or containers are NOT expected.

Any of these means you must not grade the exercise in question:

- A function is implemented in a header file (except for template functions).
- A Makefile compiles without the required flags and/or another compiler than c++.

Any of these means that you must flag the project with "Forbidden Function":

- Use of a "C" function (*alloc, *printf, free).
- Use of a function not allowed in the exercise guidelines.
- Use of "using namespace <ns_name>" or the "friend" keyword.
- Use of an external library, or features from versions other than C++98.

Yes No

Exercise 00 : Megaphone

This exercise is a warm-up to discover I/O in C++.

Operation

The goal is to develop a to_upper with specific behavior if launched without parameters. It must be solved with a C++ approach (string/upper).

Yes No

Exercise 01 : My Awesome Phonebook

This exercise is a first approach to writing a simple class and using it in an interactive program. If the exercise does not work perfectly, rate what can be graded.

Error management

There is some error management to do in this program but the behaviors are not specified in the subject. Exiting the program properly or managing errors is ok. A segfault is not! :D

Go to homepage 🏠 (../../)

Rate it from 0 (failed) through 5 (excellent)

The EXIT command

Write down the EXIT command based on what is in the subject line.

Yes No

Visibility

Contact class attributes must be private. The class must expose the corresponding accessors. Also check that anything used only in one class (and not just the Contact class) is private and the rest public. Beginners tend to to make everything public. That's what you need to check here.

Rate it from 0 (failed) through 5 (excellent)

Contact and Phonebook Classes

The code must include a Contact class (or other name). The class must contain the requested attributes. The code must contain a Phonebook class with an array of Contacts inside.

Yes No

Read/Eval Loop

The program should offer a sort of read/eval loop: Read then process input, then wait for new input until receiving an EXIT command. The loop should be done in a C++ way (std::cin).

Yes No

The ADD command

Make a note of the ADD command based on what is in the subject line.

Rate it from 0 (failed) through 5 (excellent)

Go to homepage 🏠(../../)

The SEARCH command

Write down the SEARCH command based on what is requested in the subject line. A slight deviation from the expected format is not important. This part is about using "iomanips" in C++ and that's what you need to focus on.

Rate it from 0 (failed) through 5 (excellent)

Exercise 02 : The Job of Your Dreams

The aim of this exercise is to recover information hidden amidst the noise and insert code into an existing context.

Did you save the world?

This exercise is pretty straightforward. Either Account.cpp works, or it doesn't or it doesn't. Compare the program's output with the log file provided. Any difference (except for timestamps or destructor order) means that the exercise is incorrect.

Yes No

Ratings

Don’t forget to check the flag corresponding to the defense

Ok Outstanding project

Empty work Incomplete work W Invalid compilation Cheat Crash

Concerning situation Leaks I Forbidden function Can’t support / explain code

Conclusion

Give this repository a star. ⭐

(https://github.com/rphlr/42-Evals)