

more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the benefits of the entire community.

SCALE FOR PROJECT CPP MODULE 00

(/PROJECTS/CPP-MODULE-00)

You should evaluate 1 student in this team



Git repository

Introduction

- Only grade the work that is in the student or group's Git repository.
- Double-check that the Git repository belongs to the student or the group. Ensure that the work is for the relevant project and also check that "git clone" is used in an empty folder.
- Check carefully that no malicious aliases were used to fool you and make you evaluate something other than the content of the official repository.
- To avoid any surprises, carefully check that both the evaluator and the evaluated students have reviewed the possible scripts used to facilitate the grading.
- If the evaluating student has not completed that particular project yet, it is mandatory for this student to read the entire subject prior to starting the defence.
- Use the flags available on this scale to signal an empty repository, non-functioning program, a norm error, cheating etc. In these cases, the grading is over and the final grade is 0 (or -42 in case of cheating). However, with the exception of cheating, you are encouraged to continue to discuss your work (even if you have not finished it) in order to identify any issues that may have

future.

- Remember that for the duration of the defence, no segfault, no other unexpected, premature, uncontrolled or unexpected termination of the program, else the final grade is 0. Use the appropriate flag.

You should never have to edit any file except the configuration file if it exists. If you want to edit a file, take the time to explicit the reasons with the evaluated student and make sure both of you are okay with this.

- You must also verify the absence of memory leaks. Any memory allocated on the heap must be properly freed before the end of execution.

You are allowed to use any of the different tools available on the computer, such as leaks, valgrind, or e_fence. In case of memory leaks, tick the appropriate flag.

Disclaimer

Please respect the following rules:

- Remain polite, courteous, respectful and constructively throughout the evaluation process. The well-being of the community depends on it.
- Identify with the person (or the group) evaluated the eventual dysfunctions of the work. Take the time to discuss and debate the problems you have identified.
- You must consider that there might be some difference in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade him/her as honestly as possible. The pedagogy is valid only and only if peer-evaluation is conducted seriously.

Guidelines

You must compile with clang++, with -Wall -Wextra -Werror

As a reminder, this project is in C++98 and C++20 members 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 (except in a template)
- A Makefile compiles without flags and/or with something other than clang++

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

- Use of a function not allowed in the subject
 - Use of "using namespace" or "friend"
 - Use of an external library, or C++20 features
-

Attachments

- ☐ subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/13157/en.subject.pdf>)
- ☐ Account.class.hpp (/uploads/document/document/2076/Account.class.hpp)
- ☐ 19920104_091532.log (/uploads/document/document/2077/19920104_091532.log)
- ☐ tests.cpp (/uploads/document/document/2078/tests.cpp)

Ex00: Megaphone

This exercise is a warm-up intended to discover basic C++ i/o streams.

Is it working?

This exercise is basically a to_upper program with a specific behavior when run without any parameter. Accept C approach (loop/check) and C++ approach (strings/upper).

☐ Yes

☐ No

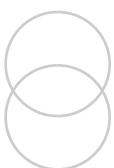
Ex01: My Awesome Phonebook

This exercise is a first approach to writing a simple class and a small interactive program that uses it. If the exercise is not fully fonctionnal, grade what can be graded.

Error handling

Some error handling is required in this exercise but behaviors are not specified in the subject.

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



Rate the EXIT command as described in the subject.

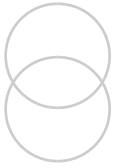
☐ Yes

☐ No

Visibility

The attributes of the class Contact should be private and the class should expose the corresponding accessors.\r\nAlso check that anything that should always be used inside a class (not only in the Contact class) is private and that anything that can be used outside a class is public. Beginners tend to put everything in public, that's what you want to spot here.

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



The Contact class

The code must include a Contact class, or whatever name the student used.

This class must contain attributes for the different fields.

☐ Yes

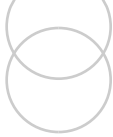
☐ No

Read/Eval loop

The program must present a read/eval loop at some point: reading the input, evaluating it, then loop until an EXIT command appears in input.

☐ Yes

☐ No



The ADD command

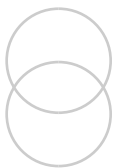
Rate the ADD command as described in the subject. The field names are not relevant, nor the number of fields as long as it respects the spirit of the exercise.

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

The SEARCH command

Rate the SEARCH command as described in the subject. A minor divergence in the expected formatting is not relevant. The aim of this part is to use C++ iomanip and that's what you should focus on.

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



Ex02: The Job Of Your Dreams

This exercise is intended to extract information and directions from useless noise, and to insert new code into an existing context.

Did you save the day?

This exercise is pretty straight forward. Either Account.class.cpp works, either it does not. Compare the program's output and the provided log. Any difference apart from the timestamp is a hint that something went horribly wrong.

☐ Yes

☐ No

Ratings

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

☐ Ok

☐ Outstanding project

☐ Empty work

☐ No author file

☐ Invalid compilation

☐ Norme

☐ Cheat

☐ Crash



☐ Leaks

1 Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

General term of use of the site (<https://signin.intra.42.fr/legal/terms/6>)

Privacy policy (<https://signin.intra.42.fr/legal/terms/5>)

Legal notices (<https://signin.intra.42.fr/legal/terms/3>)

Declaration on the use of cookies (<https://signin.intra.42.fr/legal/terms/2>)

Rules of procedure (<https://signin.intra.42.fr/legal/terms/4>)

Terms of use for video surveillance (<https://signin.intra.42.fr/legal/terms/1>)