

SCALE FOR PROJECT PISCINE CPP (/PROJECTS/PISCINE-CPP) / DAY 06 (/PROJECTS/42-PISCINE-C-FORMATION-PISCINE-CPP-DAY-06)

Introduction

The subject of this project is rather vague and leaves a lot to the user's choice. This is INTENDED. The questions in this grading scale, however, are very focused and concentrate on what we think is the core of each exercise, what we want you to grasp. So we would like you to do the same : You can and should tolerate moderate deviations in filenames, function names, etc ... as long as the exercise basically works as intended. Of course, in case the student you are grading really strayed too far, you should not grade the exercise in question at all. We leave it to your good judgement to determine what constitutes "straying too far".

Guidelines

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

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 Cheat:

- Use of a "C" function (*alloc, *printf, free)
- Use of a function not allowed in the subject
- Use of "using namespace" or "friend" (Unless explicitly allowed in the subject)
- Use of an external library, or C++11 features (Unless explicitly allowed in the subject)
- Use of "C" legacy cast

Attachments

 Subject (/uploads/document/document/1045/d06.en.pdf)

Exercise 00: Scalar conversion

This exercise shows usage of the static_cast.

Scalar conversion

Is the program running as intended, and did the student used the `static_cast` to converts values ? We'll accept the use of implicit casts for promotion casts only.

Anyway, please don't be too much uncompromising towards the exercice's outputs if the spirit of the exercice is respected. Failing this exercice does not prevent from grading the next ones. I know this exercice can be cryptic to some students.

☒ Yes

☐ No

Exercise 01: Serialization

This exercice shows usage of the `reinterpret_cast`.

Retrying of raw data

Is the program running as intended, and did the student used the `reinterpret_cast` to retype the raw data ? Remember that the raw data pointed by the `void *` must be 20 bytes (8+4+8) contiguous on the heap.

☒ Yes

☐ No

Exercise 02: Identify real type

This exercice shows usage of the `dynamic_cast`.

Real type identification

Is the program running as intended, and did the student used the `dynamic_casts` to identify the real type ? (In case you're wondering, the header must not appear anywhere.)

☒ Yes

☐ No

Ratings

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

☒ Ok

☐ Empty work

☐ Incomplete work

☐ No author file

☐ Invalid compilation

Conclusion

Leave a comment on this evaluation

Preview!!!