

Homepage (../..)

-GitHub (<https://github.com/KanekiEzz/>)

SCALE FOR PROJECT CPP MODULE 02 (HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/CPP-MODULE-02)

You should evaluate 1 student in this team

Introduction

Please respect the following rules:

- Remain polite, courteous, respectful, and constructive during the evaluation process. The well-being of the community depends on this.
- Identify with the person being assessed or the group being assessed any potential malfunctions in their work. Take the time to discuss and debate the problems identified.
- You should be aware that there may be slight differences in interpretation between the project's instructions, its scope, and its features. Keep an open mind and rate as honestly as possible. Pedagogy is only valid if peer review is done seriously.

Guidelines

- Only note what is contained in the student's or group's cloned Git repository.
- Check that the Git repository belongs to the student or group, that the project is the one expected, and that "git clone" is used in an empty folder.
- Check carefully that no alias has been used to deceive you and make sure that you are evaluating the official rendering.
- To avoid any surprises, check with the student or group about potential scripts used to facilitate the assessment (e.g., test or automation scripts).
- If you have not done the project you are going to evaluate, you must read the entire topic before starting the evaluation.
- Use the available flags to report an empty rendering, a program not working, a Standard error, cheating... In these

situations, the assessment is over and the grade is 0, or -42 in case of cheating. However, except in cases of cheating, you are encouraged to

continue to improve the work submitted, even if it is incomplete. This is to identify errors that should not be repeated in the future.

- If the subject requires a configuration file, you should never have to edit it.

If you want to edit a file, take the time to explain why to the person being assessed and make sure you have their consent.

- You should also check for memory leaks. Any memory allocated on the heap must be freed cleanly before the program finishes executing.

You are free to use any tool available on the machine, such as leaks, valgrind, or e_fence. If memory leaks occur, check the appropriate flag.

Attachments

subject.pdf (<https://github.com/rphlr/42-Subjects/>)

Preliminary tests

If cheating is suspected, grading and review will end immediately. To report cheating, select the "Cheat" flag. Use this flag calmly, carefully, and with discretion.

Prerequisites

The code must compile with c++ and the flags -Wall -Wextra -Werror. As a reminder, this project must follow the C++98 standard. Therefore, C++11 (or other standard) functions and containers are NOT expected.

Do not rate the exercise if you find:

- A function implemented in a header file (except for template functions). A Makefile that
- compiles without the requested flags and/or with something other than C++.

Select the "Forbidden function" flag if you encounter:

- Using a "C" function (*alloc, *printf, free). Using a
- function prohibited in the project.
- Using "using namespace <ns_name>" or the keyword "friend".
- Using an external library, or features specific to versions after C++98.

Yes

No

Exercise 00: My first cannon

This exercise introduces the notion of canonical class with a simple arithmetic exercise: fixed-point numbers.

Homepage (../..)

-GitHub (https://github.com/KanekiEzz/)

Makefile

There is a Makefile that compiles using the appropriate flags.

Yes

No

Accessors

The Fixed (or other) class must have accessors for the raw value:

- `int getRawBits(void) const;`
- `void setRawBits(int const raw);` Are these members present and functional?

Yes

No

Canonical class

A canonical class must have at least:

- A default constructor A
- destructor
- A copy builder
- An assignment operator Are these elements present and functional?

Yes

No

Exercise 01: First steps towards a useful classroom

The previous exercise was a good first step. However, the class was of little use since it only allowed the value 0.0 to be represented.

Makefile

There is a Makefile that compiles using the appropriate flags.

Yes

No

Constructor via floating

Can you construct an instance from a floating point number?

Yes

No

operator <<

Is there a << operator and is it functional?

Homepage (../..)/Yes

-GitHub (<https://github.com/KanekiEzz/>)

Fixed point to integer

The class must include a member function "int toInt(void) const;" that converts a fixed-point number to an int.

Is it present and functional?

Yes

No

Fixed point to float

The class must include a member function "float toFloat(void) const;" that converts a fixed-point number to a float.

Is it present and functional?

Yes

No

Construction with an int

Can we instantiate the class with the constructor taking an int?

Yes

No

Exercise 02: Now we can talk

This exercise adds comparison and arithmetic operators to the class.

Makefile

There is a Makefile that compiles using the appropriate flags.

Yes

No

Comparison operators

Are the 6 comparison operators (>, <, >=, <=, == and !=) present and functional?

Yes

No

Arithmetic operators

Are the 4 arithmetic operators (+, -, * and /) present and working? (If you perform a division by 0, it is acceptable for the program to crash.)

Yes

No

Homepage (../..)

-GitHub (<https://github.com/KanekiEzz/>)**Other operators**

Are the four increment and decrement operators (pre-increment and post-increment, pre-decrement and post-decrement) present and functional?

Yes

No

Overloading public static member functions

Finally, check that the static member functions min() and max() are implemented and working.

Yes

No

Exercise 03: BSP

This exercise should make you realize how easy it is to implement complex algorithms once you get the basics working as expected.

Makefile

There is a Makefile that compiles using the appropriate flags.

Yes

No

Point Class

There is a Point class which has two attributes (x and y) of type Fixed const.

It also has a constructor that takes two floats and initializes x and y with those values.

Yes

No

bsp function

There is a bsp() function whose prototype is

"bool bsp(Point const a, Point const b, Point const c, Point const point)".

The function returns True if the point is inside the triangle described by vertices a, b, and c. Otherwise, it returns False.

Yes

No

Main and tests

There is a main to test that the bsp() function works as described above. Run several tests to ensure that the value of re ecr

Hsotmoerpeacgte tower. (../..)

-GitHub (<https://github.com/KanekiEzz/>)

Yes

No

Ratings

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

Ok

Outstanding project

Empty work

Incomplete work

WInvalid compilation

Cheat

Crash

Concerning the situation

Leaks

LForbidden function

Can't support / explain code

Conclusion

Give this repository a star.☆

(https://github.com/KanekiEzz/1337_cursus_42)

Follow @KanekiEzz on GitHub

(<https://github.com/KanekiEzz>)