

SCALE FOR PROJECT CPP MODULE 02 ([HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/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 rests on this.
- Identify with the person being evaluated or the group being evaluated any possible dysfunctions in their work. Take the time to discuss it and debate the problems identified.
- You should take into account that there may be slight differences in interpretation between the project instructions, its scope and its functionalities. Keep an open mind and grade as honestly as possible. The pedagogy is only valid if the peer-evaluation 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 expected one, and that "git clone" is used in an empty folder.
- Carefully check that no alias has been used to mislead you and make sure that you evaluate the official rendering correctly.
- To avoid any surprises, check with the student or group about potential scripts used to facilitate the assessment (for example, 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 evaluation is finished and the score is 0, or -42 in the event of cheating . However, with the exception of cases of cheating, you are encouraged to continue the discussion on the work submitted, even if it is incomplete. This is to identify errors that should not be reproduced in the future.

- If the topic requires a configuration file, you should never have to modify it. If you wish to edit a file, take the time to explain why to the person being evaluated and ensure that you have their consent.

- You must also check for memory leaks. Any memory allocated on the heap must be freed cleanly before program execution ends . You have the right to use any tool available on the machine such as leaks, valgrind or e_fence. In case of memory leaks, check the appropriate flag.

Attachments

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

Preliminary tests

If cheating is suspected, grading and evaluation ends immediately. To report this, select the "Cheat" flag. Be careful to use it calmly, carefully and with discernment.

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 grade the exercise if you find:

- Une fonction implémentée dans un fichier d'en-tête (sauf pour les fonctions templates).
- Un Makefile compilant sans les flags demandés et/ou avec autre chose que c++.

Sélectionnez le flag "Fonction interdite" (Forbidden function) si vous rencontrez :

- L'utilisation d'une fonction "C" (*alloc, *printf, free).

- L'utilisation d'une fonction interdite dans le projet.
- L'utilisation de "using namespace <ns_name>" ou du mot-clé "friend".
- L'utilisation d'une bibliothèque externe, ou de fonctionnalités propres aux versions postérieures à C++98.

Yes

No

Exercice 00 : Mon premier canon

Cet exercice introduit la notion de classe canonique avec un exercice arithmétique simple : les nombres à point fixe.

Makefile

Il y a un Makefile qui compile en utilisant les flags appropriés.

Yes

No

Accesseurs

La classe Fixed (ou autre) doit avoir des accesseurs pour la valeur brute :

- `int getRawBits(void) const;`
- `void setRawBits(int const raw);` Ces membres sont-ils présents et fonctionnels ?

Yes

No

Classe canonique

Une classe canonique doit avoir au moins :

- Un constructeur par défaut
- Un destructeur
- Un constructeur par copie
- Un opérateur d'affectation Ces éléments sont-ils présents et fonctionnels ?

Yes

No

Exercice 01 : Premiers pas vers une classe utile

L'exercice précédent était un bon premier pas. Cependant, la classe était peu utile puisqu'elle ne permettait de représenter que la valeur 0.0

Makefile

Il y a un Makefile qui compile en utilisant les flags appropriés.

Yes

No

Constructeur via flottant

Peut-on construire une instance à partir d'un nombre flottant ?

Yes

No

opérateur <<

Y a-t-il un opérateur << et est-il fonctionnel ?

Yes

No

Point fixe vers entier

La classe doit inclure une fonction membre "int toInt(void) const;" qui convertit un nombre à point fixe en int.
Est-elle présente et fonctionnelle ?

Yes

No

Point fixe vers float

La classe doit inclure une fonction membre "float toFloat(void) const;" qui convertit un nombre à point fixe vers un float.
Est-elle présente et fonctionnelle ?

Yes

No

Construction avec un int

Peut-on instancier la classe avec le constructeur prenant un int ?

Yes

No

Exercice 02 : Maintenant, on peut parler

Cet exercice ajoute les opérateurs de comparaison et arithmétiques à la classe.

Makefile

Il y a un Makefile qui compile en utilisant les flags appropriés.

Yes

No

Opérateurs de comparaison

Les 6 opérateurs de comparaison (>, <, >=, <=, == et !=) sont-ils présents et fonctionnels ?

Yes

No

Opérateurs arithmétiques

Les 4 opérateurs arithmétiques (+, -, * et /) sont-ils présents et fonctionnels ?
(Si vous effectuez une division par 0, il est acceptable que le programme crash.)

Yes

No

Autres opérateurs

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, verify that the static member functions min() and max()
are implemented and functional.

Yes

No

Exercise 03: BSP

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

Makefile

There is a Makefile that compiles using the appropriate flags.

Yes

No

Class Point

There is a Point class that 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 function bsp() 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 the return value is correct.

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

! Forbidden function

Can't support / explain code

Give this repository a star. ★

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