

(https://profile.intra.42.fr)

방어의 질, 따라서 노동 시장에서 학교의 질은 당신에게 달려 있다는 것을 기억하십시오. Covid 위기 동안 원격 방어는 더 많은 유연성을 허용하여 커리큘럼으로 진행할 수 있지만 모든 사람의 기술 개발에 해를 끼치는 사기, 불의, 게으름의 위험을 더 많이 가져옵니다. 우리는 전체 커뮤니티의 이익을 위해 이러한 원격 방어 중 귀하의 성숙함과 지혜를 믿습니다.

프로젝트 CPP 모듈 02의 (/PROJECTS/CPP-MODULE-02) 규모 (/PROJECTS/CPP-MODULE-02)

이 팀에서 1명의 학생을 평가해야 합니다.



Git 저장소

git@vogosphere.42seoul.kr:vogosphere/intra-uuid-27e701d1-6d



소개

- 학생 또는 그룹의

Git 저장소 에 있는 작업만 채점 합니다.

- Git 저장소가 학생

또는 그룹에 속하는지 다시 확인합니다 . 해당 프로젝트에 대한 작업인지 확인하고 빈 폴더에 "git clone"이 사용되었는지도 확인합니다.

- 사용자를 속이고 공식 저장소

의 내용이 아닌 다른 것을 평가하도록 악의적인 별칭이 사용되지 않았는지 주의 깊게 확인하십시오

- 놀라움을 피하기 위해

평가 대상 학생과 평가 대상 학생 모두

가 채점을 용이하게 하는 데 사용된 가능한 스크립트를 검토했는지 주의 깊게 확인합니다 .

- 평가하는 학생이 특정

프로젝트를 아직 완료하지 않은 경우 이 학생은 반드시 읽어야 합니다.

방어를 시작하기 전에 전체 주제.

- 이 척도에서 사용 가능한 플래그를 사용하여 빈 저장소, 작동

하지 않는 프로그램, 표준 오류, 부정 행위 등을 알립니다. 이러한 경우

등급이 종료되고 최종 등급은 0(또는

부정 행위의 경우 -42)입니다. 그러나 부정 행위를 제외하고 이 실패의 원인 이 될 수 있는 문제를 식별 하고 앞으로 같은 실수를 반복하지 않도록

작업을 계속 논의 하는 것이 좋습니다(작업을

완료 하지 않았더라도)

.

- 방어 기간 동안 세그폴트,
기타 예상치 못한, 시기상조, 통제할 수 없는 또는 예기치 않은
프로그램 종료 가 없으며 , 그렇지 않으면 최종 등급은 0입니다.
적절한 플래그.
존재하는 경우 구성 파일을 제외한 모든 파일을 편집할 필요가 없습니다.
파일을 편집하려면 시간을
내어 평가 대상 학생과 함께 이유를 밝히고 두 사람 모두 괜찮은지 확인하십시오.

- 또한 메모리 누수가 없는지 확인해야 합니다. 힙에 할당된 모든 메모리
는 실행이 끝나기 전에 적절하게 해제 되어야 합니다 . 누출, valgrind 또는 e_fence
와 같이 컴퓨터에서 사용할 수 있는 다양한 도구를 사용할 수 있습니다
. 메모리 누수의 경우 해당 플래그를 선택하십시오.

부인 성명

다음 규칙을 준수하십시오.

-

평가 과정 전반에 걸쳐 공손하고, 예의 바르고, 존중하고, 건설적인 태도를 유지하십시오. 커뮤니티의 웰빙은
그것에 달려 있습니다.

-

작업 의 궁극적인 기능 장애를 평가한 사람(또는 그룹)과 동일시합니다 . 시간을
내어 여러분이 식별한 문제 에 대해 토론 하고 토론하십시오.

-

동료들이 프로젝트의 지침과
기능 범위를 이해하는 방식에 약간의 차이가 있을 수 있음을 고려해야 합니다 . 항상 열린 마음을 유지하고
가능한 한 정직하게 평가하십시오. 교수법은
동료 평가가 진지하게 수행되는 경우에만 유효합니다 .

지침

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 "C" function (*alloc, *printf, free)
- Use of a function not allowed in the subject
- Use of "using namespace" or "friend"
- Use of an external library, or C++20 features
- Use of an already existing container, or any existing function, to implement another container

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/26280/en.subject.pdf>)

Exercise 00: My First Canonical

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

Makefile

There is a Makefile that compiles using the appropriate flags.

☒ Yes

☐ No

Accessors

The Fixed class (or whatever its name) must provide accessors to the raw value:

```
- int getRawBits( void ) const;  
- void setRawBits(  
int const raw );
```

Are these member functions present and functional?

☒ Yes

☐ No

Canonical

A canonical class must provide at least:

- A default constructor
- A destructor
- A copy constructor
- An assignation operator

Are these elements present and functional?

☒ Yes

☐ No

Exercise 01: Towards a more useful fixed point class

Ex00 was a good start, but our class is still pretty useless being only able to represent the fixed point value 0.0.

Makefile

There is a Makefile that compiles using the appropriate flags.

☒ Yes

☐ No

Floating point constructor

Is it possible to construct an instance from a floating point value?

☒ Yes☐ No

<< operator

Is there a << operator overload and is it functional?

☒ Yes☐ No

Fixed point value to integer value

A member function "int toInt(void) const;" that converts the fixed point value to an integer value must be present. Is it functional?

☒ Yes☐ No

Fixed point value to floating point value

A member function \ "float toFloat(void) const;\" that converts the fixed point value to a float value must be present. Is it functional?

☒ Yes☐ No

Integer constructor

Is it possible to construct an instance from an integer value?

☒ Yes☐ No

Exercise 02: Now we're talking

This exercise add comparison and arithmetic features to the class.

Makefile

There is a Makefile that compiles using the appropriate flags.

☒ Yes☐ No

Division

The class must provide a division operator. Is it present and functional?

☒ Yes☐ No

Exercise 03: BSP

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

Makefile

There is a Makefile that compiles using the appropriate flags.

☒ Yes

☐ No

Class Point

There is a class Point which has two attributes (x and y) of type Fixed const.
It also has a constructor that takes two floats and initialize x and y with those values.

☒ Yes

☐ No

Function bsp

There is a function bsp which 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 of the triangle described by the vertices a, b, and c.
It returns False otherwise.

☒ Yes

☐ No

Main and tests

There is a main to test that the function bsp works as metined above.
Run several test to make sure that the return value is correct.

☒ Yes

☐ No

Ratings

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

☒ Ok

☐ Empty work

☐ Incomplete work

☐ Invalid compilation

☐ Cheat

☐ Crash

☐ Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

terms & conditions (<https://signin.intra.42.fr/legal>)