МАСШТАБ ДЛЯ ПРОЕКТА CPP MODULE 07 (/PROJECTS/CPP-MODULE-07)

Вы должны оценить 1 студента в этой команде

★

Git-репозиторий

```
git@vogsphere.kzn.21-school.ru:vogsphere/intra-uuid-71018
```

Введение

Please comply with the following rules:

- Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it.

- Identify with the student or group whose work is evaluated the possible dysfunctions in their project. Take the time to discuss and debate the problems that may have been identified.

- You must consider that there might be some differences in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade them as honestly as possible. The pedagogy is useful only and only if the peer-evaluation is done seriously.

Guidelines

- Only grade the work that was turned in the Git repository of the evaluated student or group.

- Double-check that the Git repository belongs to the student(s). Ensure that the project is the one expected. Also, check that 'git clone' is used in an empty folder.

- Check carefully that no malicious aliases was used to fool you and make you evaluate something that is not the content of the official repository.

- To avoid any surprises and if applicable, review together any scripts used to facilitate the grading (scripts for testing or automation).

- If you have not completed the assignment you are going to evaluate, you have

to read the entire subject prior to starting the evaluation process.

- Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, and so forth.
In these cases, the evaluation process ends and the final grade is 0, or -42 in case of cheating. However, except for cheating, student are strongly encouraged to review together the work that was turned in, in order to identify any mistakes that shouldn't be repeated in the future.

- 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.

Attachments

ex00.cpp (/uploads/document/document/11109/ex00.cpp)

ex01.cpp (/uploads/document/document/11110/ex01.cpp)

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/61100/en.subject.pdf)

main.cpp (/uploads/document/document/11111/main.cpp)

Preliminary tests

If cheating is suspected, the evaluation stops here. Use the "Cheat" flag to report it. Take this decision calmly, wisely, and please, use this button with caution.

Prerequisites

The code must compile with c++ and the flags -Wall -Wextra -Werror
Don't forget this project has to follow the C++98 standard. Thus, C++11 (and later) 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 file (except for template functions).
- A Makefile compiles without the required flags and/or another compiler than c++.

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

- Use of a "C" function (*alloc, *printf, free).
- Use of a function not allowed in the exercise guidelines.
- Use of "using namespace <ns_name>" or the "friend" keyword.
- Use of an external library, or features from versions other than C++98.

Yes No

Exercise 00: Start with a few functions

This exercise is about writing 3 simple function templates: swap(), min() and max().

Simple types

Refer to the subject for the expected output with simple types, such as int.

Yes No

Complex types

Функции также работают со сложными типами? (проверьте файл ex00.cpp во вложении)

Да Нет

Упражнение 01: Итер

Это упражнение посвящено написанию универсальной функции для перебора массивов.

Это работает?

Протестируйте код ex01.cpp (в приложениях) с помощью iter оцениваемого учащегося.
Если все прошло хорошо, должно появиться:

```
0
1
2
3
4
42
42
42
42
42
```

Да Нет

Упражнение 02. Массив

Это упражнение посвящено написанию шаблона класса, который ведет себя как массив. Если внутреннее выделение фактического массива происходит не из-за использования new[], не оценивайте это упражнение. Попросите оцениваемого учащегося доказать, что шаблон класса работает с массивами как простых, так и сложных типов, прежде чем оценивать упражнение.

Конструкторы

Можно ли создать пустой массив и массив определенного размера?

☑ Да ☒ Нет

Доступ

Элементы должны быть доступны для чтения и записи через оператор []
(или только для чтения, если экземпляр является константным). Доступ к элементу, находящемуся вне допустимого диапазона, должен вызвать исключение std::exception.

☑ Да ☒ Нет

Рейтинги

Не забудьте проверить флаг, соответствующий защите

✔ Ok ★ Выдающийся проект

Пустая работа | Незавершенная работа | Недопустимая компиляция | Изменять

Крушение | Относительно ситуации | Утечки | Запрещенная функция

Вывод

Оставьте комментарий к этой оценке

Завершить оценку