

peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade him/her as honestly as possible. The pedagogy is valid only and only if peer-evaluation is conducted seriously.

Guidelines

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

Attachments

 [subject.pdf](#)

ex00

As usual, there has to be a main function that contains enough tests to prove the program works as required. If there isn't, do not grade this exercise. If any non-interface class is not in Coplien's form, do not grade this exercise.

ex00

There is a templated function easyFind(T, int) that does what the subject requires.

It HAS to use STL algorithms.

If it does not (Manual search using iterators for example), count it as wrong.

☐ No

ex01

As usual, there has to be a main function that contains enough tests to prove the program works as required. If there isn't, do not grade this exercise. If any non-interface class is not in Coplien's form, do not grade this exercise.

ex01

There is a class that respects the constraints of the subject.

Its member functions use STL algorithms to find their result, as much as possible.

☐ No

Better addNumber

There's a way to add numbers that's more practical than calling addNumber repeatedly.

☐ No

ex02

As usual, there has to be a main function that contains enough tests to prove the program works as required. If there isn't, do not grade this exercise. If any non-interface class is not in Coplien's form, do not grade this exercise.

ex02

There is a MutantStack class that inherits from std::stack (Or is composed of one, dealer's choice), and offers all of its member functions.

It has an iterator, and it is possible to do at least the operations in the subject's example with it.

☐ No

Better tests

There is a test main() function that has more tests than the one in the subject.

☐ No

Ratings

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

Conclusion

Leave a comment on this evaluation