42 Evals

Back to all evaluation sheets

**Points earned**
**0**

# CPP Module 07

You should evaluate 1 student in this team

## Introduction

Please follow the rules below:

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

✓ Work with the student or group being evaluated to identify potential issues in their project. Take time to discuss and debate the problems identified.

✓ Understand that there may be differences in how peers interpret the project instructions and scope. Always keep an open mind and grade as honestly as possible. Pedagogy is effective only when peer evaluations are taken seriously.

## Guidelines

Please follow the guidelines below:

✓ Only grade the work submitted to the Git repository of the evaluated student or group.

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

✓ Carefully verify that no malicious aliases are used to d
grading non-official content.

✓ If applicable, review any scripts used for testing or aut
student.

**Points earned**
0

✓ If you haven't completed the assignment you're evaluating, read the entire subject before starting the evaluation.

✓ Use the available flags to report an empty repository, a non-functioning program, a Norm error, or cheating. The evaluation process ends with a final grade of 0 (or -42 for cheating). However, except in cases of cheating, students are encouraged to review the work together to identify mistakes to avoid in the future.

✓ Remember that no segfaults or other unexpected program terminations will be tolerated during the evaluation. If this occurs, the final grade is 0. Use the appropriate flag.

✓ You should not need to edit any files except the configuration file, if it exists. If editing a file is necessary, explain the reasons to the evaluated student and ensure mutual agreement.

✓ Verify the absence of memory leaks. All memory allocated on the heap must be properly freed before the program ends.

✓ You may use tools like leaks, valgrind, or e_fence to check for memory leaks. If memory leaks are found, tick the appropriate flag.

## Attachments

Please download the attachments below:

subject.pdf

ex00.cpp

ex01.cpp

main.cpp

Points earned
0

## Mandatory Part

### 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

Points earned

0

Yes No

## Complex types

Complex types

Do the functions also work with complex types? (check with the ex00.cpp file in attachment)

Yes No

## Exercise 01: Iter

This exercise is about writing a generic function to iterate through arrays.

*Does it work?*

Test the code ex01.cpp (in attachments) with the iter of the evaluated student.

If everything went well, it should display:

```bash

0

1
```

2

3

4

42

42

42

42

42

`

**Points earned**

**0**

Yes No

## Exercise 02: Array

This exercise is about writing a class template that behaves like an array. If the inner allocation of the actual array does not come from a use of new[], don't grade this exercise. Ask the evaluated student to prove the class template works with arrays of both simple and complex types before grading the exercise.

*Constructors*

Is it possible to create an empty array and an array of a specific size?

Yes No

### Access

Access

Elements must be accessible for reading and writing thr
reading only if the instance is const). Access to an elem
must throw an std::exception.

**Points earned**
**0**

Yes No

## Bonus Part

### no bonus

no bonus

no bonus

Yes No

## Ratings

✓ OK ☆ Outstanding Empty Work

Incomplete Work Invalid Compilation

Cheat Crash Conc

Leaks Forbidden Functions

**Points earned**

**0**