



# C++ - Module 09

## STL

*Summary:*

*This document contains the exercises of Module 09 from C++ modules.*

*Version: 2.2*

# Contents

<b>I</b>	<b>Introduction</b>	<b>2</b>
<b>II</b>	<b>General rules</b>	<b>3</b>
<b>III</b>	<b>Module-specific rules</b>	<b>5</b>
<b>IV</b>	<b>Exercise 00: Bitcoin Exchange</b>	<b>6</b>
<b>V</b>	<b>Exercise 01: Reverse Polish Notation</b>	<b>8</b>
<b>VI</b>	<b>Exercise 02: PmergeMe</b>	<b>10</b>
<b>VII</b>	<b>Submission and peer-evaluation</b>	<b>13</b>

# Chapter I

## Introduction

*C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes" (source: [Wikipedia](#)).*

The goal of these modules is to introduce you to **Object-Oriented Programming**. This will be the starting point of your C++ journey. Many languages are recommended to learn OOP. We decided to choose C++ since it's derived from your old friend C. Because this is a complex language, and in order to keep things simple, your code will comply with the C++98 standard.

We are aware modern C++ is way different in a lot of aspects. So if you want to become a proficient C++ developer, it's up to you to go further after the 42 Common Core!

# Chapter II

## General rules

### Compiling

- Compile your code with `c++` and the flags `-Wall -Wextra -Werror`
- Your code should still compile if you add the flag `-std=c++98`

### Formatting and naming conventions

- The exercise directories will be named this way: `ex00`, `ex01`, ... , `exn`
- Name your files, classes, functions, member functions and attributes as required in the guidelines.
- Write class names in **UpperCamelCase** format. Files containing class code will always be named according to the class name. For instance:  
`ClassName.hpp/ClassName.h`, `ClassName.cpp`, or `ClassName.hpp`. Then, if you have a header file containing the definition of a class "BrickWall" standing for a brick wall, its name will be `BrickWall.hpp`.
- Unless specified otherwise, every output messages must be ended by a new-line character and displayed to the standard output.
- *Goodbye Norminette!* No coding style is enforced in the C++ modules. You can follow your favorite one. But keep in mind that a code your peer-evaluators can't understand is a code they can't grade. Do your best to write a clean and readable code.

### Allowed/Forbidden

You are not coding in C anymore. Time to C++! Therefore:

- You are allowed to use almost everything from the standard library. Thus, instead of sticking to what you already know, it would be smart to use as much as possible the C++-ish versions of the C functions you are used to.
- However, you can't use any other external library. It means C++11 (and derived forms) and Boost libraries are forbidden. The following functions are forbidden too: `*printf()`, `*alloc()` and `free()`. If you use them, your grade will be 0 and that's it.