

README

June 3, 2024

1 Functional programming in C++

Functional programming means coming back closer to what **variables** and **functions** are in **mathematics**: the former, immutable data, and the latter, transformations from an input into an output.

In this presentation, we will see how to modify our C++ programs to approach the functional paradigm, by the way formalizing already known good practices, and facilitating parallel and distributed execution.

1.1 To be continued

1. [Adopt a functional style in old C ++.](#)
2. [Lambda functions.](#)
3. [Higher-order functions.](#)
4. [A brief introduction to the algebra of types.](#)
5. [Towards monads.](#)
6. [C++20 functional pipes with Ranges](#)

1.2 Inspirations

- John Carmack & Arzar : [Programmation fonctionnelle en C++](#)
- Ivan Cukic : [Functional Programming in C++](#)
- Bartosz Milewski : [Category Theory for Programmers](#)

© CNRS 2019

This document was created by David Chamont, proofread and improved by Hadrien Grasland and translated by Olga Abramkina. It is available under the [License Creative Commons - Attribution - No commercial use](#) - Shared under the conditions [4.0 International](#)