# MQF 633: C++ for Financial Engineering

### David Xue Feng, 2024

## 20 years working experience, worked in manufacturing, software, banks and fin-tech, as an engineer, consultant, developer, quant etc. Now as head of quant in Maybank.

## Lecturer Background

## Learning the journey of programming. From MATLAB, R to VBA, then C/C++, Java, C#. A late self-learner for C++. Only start to learn C++ around 2012. Now, I am quite comfortable to work in any software development domain with C++, Python, Java, C# etc

## For C++, it is a tough learning journey: from textbook, training courses, to on the job try and error, and lately from all sorts on-line resources. I am still learning new and interesting topics.

* Till today, in many areas C++ is still a very commonly used and yet challenging language. C++ itself also adapting almost every year to modern topics. But one can truly master it as long as one gets it start, with proper tools, and keep learning and improving

## 10 lectures, homework assignments (project) and final exam

## Course Structure

Basics of C++ (part 1)

* Lecture 1: Introduction of C++ basics
* Lecture 2: Data types and basic syntax
* Lecture 3: Functions and Modular Programming
* Lecture 4: Object-Oriented Programming (OOP)
* Lecture 5: Object-Oriented Programming (OOP2) with examples

Advanced topics (part2)

* Lecture 6: Standard Template Library Examples
* Lecture 7: Template and design patterns
* Lecture 8: Concurrency, multi-threading and parallelism
* Lecture 9: From C/C++ to Python
* Lecture 10: Project explanation, Q&A

Prerequisite

* Install VS Code (mac) or Visual Studio for windows
* Install GitHub and get ready to share code through Github

Goal of the course

* Level 1: Be able to understand most of C++ syntax and debug C++ program
* Level 2: Be able to understand and write a simple OOP C++ program
* Level 3, understand the usage of smart pointer, references and const etc
* Level 4: understand basic principle of multi-threading, concurrency and thread safety
* Level 5: understand the commonly used design pattern

Marking

* Attendance (20%)
* Assignments (30%)
* Project (50%)