

Syllabus

Introduction to C++ Programming and Numerical Analysis Spring 2017

Instructor: Dave Klein
Email: dave.klein.mfe@gmail.com
Class Time: Saturdays, 9 a.m. – 1 p.m.

GSI: TBD
Email: mfe.cpp@gmail.com

Overview:

This course offers an introduction to C++ programming with an emphasis on quantitative finance. It is intended for incoming MFE students and individuals who are considering applying for the program. There are no prerequisites for the course. However, students with little or no programming experience are encouraged to perform self-study ahead of time. Further, we focus on numerical analysis and quantitative finance applications (options, simulation, fixed income valuation). No previous exposure to numerical analysis or finance is required but students should at least be fluent in calculus (derivatives, integrals, etc.).

There will be a total of thirteen lectures in the course. The first lecture is intended as a quick introduction to C++ as well as to provide instruction on how to install the programming environment and debug programs. The next four lectures cover the basics of the C++ language as well as selected topics in numerical analysis and finance. The next five lectures develop concepts about object-oriented programming. The final lectures cover selected topics in quantitative finance with a focus on programming solutions in C++.

Completing coding exercises is vital to learning a new programming language. This course has five mandatory assignments that must be completed.

There are no text books for the course. Lecture notes will be made available in advance. There are no suggested additional books for the course but we do suggest bookmarking and referring to:

Cplusplus.com: <http://www.cplusplus.com> – for extensive information about the C++ language

Stack Overflow: <http://stackoverflow.com> – for general programming questions

Programming Environment:

Students are required to have access to a C++ program environment. **Instructions will be given in Assignment 1 on how to install a specific C++ development environment.** Students are allowed to use whatever C++ environment is most convenient for them and it is their responsibility to ensure they can write, save and execute programs. Often, part of an assignment submission will be multiple C++ files, so students should make sure that they can save C++ projects rather than write them solely online.

The course is designed to teach general C++ programming so a newer environment that supports C++ 11 or C++ 14 is not required. Older programming environments are fine for use.

Communication:

We will use bConnected and Study.net for communication.

For general course questions, please contact the GSI's at mfe.cpp@gmail.com. **Students are highly encouraged to use the forum in bConnected to ask and answer questions.** It has been our experience that answers are often supplied there faster than via email. Also, asking questions in the forum provides answers for other students with the same question. Please make sure not to upload solutions to homework in the forum.

Assignments:

There are 5 assignments in total and they will be made available via Study.net. It is vital that students complete the assignments. Point totals will be clearly stated on each assignment. **Students must receive at least 65% of total available points to receive the certificate for the course.**

Assignments will be submitted via bConnected. Our deadlines are strict. Assignments submitted within one day after the deadline will receive a 50% point deduction. Assignments will not be accepted 24 hours after the deadline. **No exceptions will be made.**

Students are allowed to consult outside material when doing their work. However, students may not submit the work of others as their own. Additionally, students should do their own work, including all the coding. **Students should not submit assignments together.** If we deem that students have submitted work that is not their own or was completed as part of a joint effort, points will be deducted. **If we believe plagiarism has occurred or a student has submitted work that is not their own, it is possible that we will not award the course certificate and, depending on the severity of the infraction, will notify the MFE office of the offense.**

Students are encouraged to discuss the material and assignments, especially in the forum. However, it is the responsibility of the student to complete each assignment individually. In other words, share ideas, resources and thoughts but do not share code or written answers to questions. Lively discussion in the forum is encouraged.

Posting of course material (assignments, slides, solutions, etc.) is strictly forbidden. Posting of material on the web is strictly prohibited and will lead to legal action.

Grades:

There are no grades in the course. **Students who receive at least 65% of the total available points will receive a certificate that they passed the course.** There is no “fail” grade. Students either receive the certificate or they do not.

Exam:

There is no exam in the course.

Lectures:

Lectures take place on Saturdays and are generally posted online within a few days after that. We will attempt to stick to the following schedule but may deviate occasionally. Topics may also be changed in order to better meet the needs of students.

SCHEDULE – Numerical Analysis & Finance Topics are underlined

January 14, 2016: 9 a.m. – 11 a.m.

Lecture 1: A brief history of C++, introduction to the programming environment, writing & debugging our first program, help with getting started

January 21, 2016: 9 a.m. – 1 p.m.

Assignment 1 due

Lecture 2: Basic syntax, data types, variables, operators, simple I/O, vanilla options

Lecture 3: Conditional statements, loops, functions, scope, root-finding methods

January 28, 2016: 9 a.m. – 1 p.m.

Assignment 2 due

Lecture 4: Arrays, strings, pointers, references, dynamic vs. static memory allocation

Lecture 5: Data structures, classes & objects, pseudorandom numbers, simulation

February 4, 2016: 9 a.m. – 1 p.m.

Lecture 6: OOP, public, private, inheritance

Lecture 7: Overloading, polymorphism, virtual functions, correlation, volatility

February 11, 2016: 9 a.m. – 1 p.m.

Assignment 3 due

Lecture 8: Information hiding, encapsulation, namespaces, preprocessor, breaking projects into multiple files

Lecture 9: Files & streams, bonds, credit risk

February 25, 2016: 9 a.m. – 1 p.m.

Lecture 10: Generic programming, templates, Standard Library, STL

Lecture 11: Black-Scholes, binomial trees

March 4, 2016: 9 a.m. – 1 p.m.

Assignment 4 due

Lecture 12: Bond term structure, multivariate optimization

Lecture 13: TBD

March 11, 2016:

Assignment 5 due