Course Syllabus

Object Oriented Programming in Finance I & II

Fall 2011 / Spring 2012

The goal of this year-long sequence of courses is to give a rigorous introduction to computer programming and software engineering with special emphasis on applications to financial engineering. Our primary programming language will be C++. This programming language is fast enough to accommodate the performance demanded in financial environments. At the same time C++ is an Object oriented language and, as such, is suitable for modern software design.

In this course the assumption is that students have had no background in computer programming, although even people who are familiar with some programming language will hopefully benefit and learn new material.

In part I in the Fall semester the course will start with basic concepts of programming, but we quickly get into topics in object oriented programming, UML diagrams, and basic patterns. We will also include introduction to basic algorithms and data structures.

In part II in the Spring semester, more advanced topics will be covered, including advanced algorithms and data structures especially through introduction to STL and boost libraries, numerical algorithms and introduction to BLAS and LAPACK libraries, design of graphical user interfaces, and concurrent programming (also known as multiprogramming).

Course Web site and communications:

The course web site is on http://sakai.rutgers.edu. Once logged in (using your Rutgers netID and password), you should see a tab for each course you are eligible, including this course.

You should make a habit of checking this web site at least two or three times a week. All homework assignments will be posted on Sakai, and you will be asked to submit your work on Sakai as well. Also all course hand-outs will be posted on the modules page of Sakai.

Important: When trying to contact me, please DO NOT SEND E-MAIL. Instead use Sakai's Discussions and Private messages tab. If your question or comment is not of private nature and can benefit others use the discussion board. For instance, if you find an error in posted material, or you do not understand a point, it is better to post your question on the discussion board. Then I, or another student who may know the answer, can post a reply. Every student is encouraged to take part in discussion board and answer questions if they know them, or extend and comment, when possible.

If on the other hand you have an issue that is private or not relevant to others (for instance you want to discuss your grade, or need extension, etc.) then use Sakai's private message to contact me.

Office hours

My office hours are Wednesdays from 4-5PM in WP 1062. My phone number in Newark is 973-353-5488. Other days of the week I am not in Newark. Information about my office at RUTCOR in the Busch Campus in Piscataway is posted in Sakai.

You do not need to make an appointment to see me during office hours. For all other times please send me a PM through Sakai to make an appointment. I am in Newark on Wednesdays, but other days I will am usually at RUTCOR.

Textbooks

For the Fall semester the following texts will be used, though a lot of additional material will be distributed through Sakai.

1) Required
Title: C++ for Programmers
Author: P. Dietel and H. Dietel
Publisher: Prentice Hall
Year: 2009 (or 2010)
ISBN: 0-13-700130-4
2) Required
Title: The C Programming Language Second edition (or the latest edition)
Author: B. Kernighan and D. Ritchie
Publisher Prentice Hall
Year: 1988
ISBN: 0-13-110362-8

3) Recommended
Title: Effective C++ Third Edition
Author: S. Myers
Publisher: Addison-Wesley
Year: 2005

Books for the Spring semester will be announced shortly.

Grading policy

1) There will be roughly one programming project every two weeks. These projects are absolutely essential and students cannot expect to understand these courses without making serious efforts in completing these programming projects. As a result regardless of your grades in exams and quizzes, no student will pass the course if they fail to hand in more than one programming assignment per semester. 50% of your grade is based on programming projects, I may also decide to give group projects towards the end of the semester (or may be in the next semester.)

2) There will be occasional in-class short quizzes 10%

3) Final exam 40%

Software downloads

Throughout the course you will be required to download software for various purposes. This includes compilers, IDE's UML tools, and possibly some libraries. These software are all open source and free, but you need to install and configure them on your computers. Detailed instructions will be provided in the Modules page of Sakai.

Tentative schedule and list of topics

The following is a tentative list of topics I plan to cover for the entire year. More detail, including section of texts to be read in conjunction with these topics will be posted in the modules section of Sakai, as we progress.

Fall Semester:

Week 1: A review of computer architecture and hardware, machine language, assembly language, high level programming compilers (1)

Week 2-3: Basics of C programming: types, assignments, operations, control structures, functions, the notion of structured programming, pointers and arrays, the standard C library (2.5)

Week 3-4: Onto C++, Classes and objects, member functions and variables (1.5)

Week 5-6: More on Classes, inheritance, UML Class diagrams (2)

Week 7-8: Introduction to algorithms: simple sorting and searching, UML activity diagrams and flow charts (2)

Week 9: Constructors and destructors, new() and delete() functions, garbage collection (1)

Week 10: Abstraction and virtual functions, function overloading and its application in OO programming (1)

Week 11: Introduction to basic patterns (0.5)

Week 11-12: Templates, generic programming multiple inheritance, Introduction to C++ STL (Standard Template Library) (1)

Spring Semester:
Week 1: More patterns (1)

Week 2-4: Data structures and algorithms: stacks and queues, heaps, balanced trees and hash tables, implementation in STL and Boost libraries, OO implementation (3)

Week 5-6: Numerical algorithms: polynomials, divided difference, integration, vector/matrix operations, BLAS and LAPACK libraries in Boost library with some (2)

Week 7-8: Concurrency: threads, mutual exclusion, semaphores, locking and unlocking (2)

Week 9-10: Interfaces with databases, and streams (2)

Week 11-12: Graphical user interface design (time permitting) (2)