

EECE3326, Optimization Methods
Department of Electrical and Computer Engineering
Fall 2009, MWTh 10:30am-11:35am, Snell Library 039

Syllabus

Instructor: Ningfang Mi, Assistant Professor
E-mail: ningfang@ece.neu.edu, Office: 302 Dana, Mailbox: 440 Dana
Office hours: By appointment
Teaching Assistant: TBD

CONTENT

In EECE3326 you will study a collection of classic algorithmic techniques that are useful for solving engineering problems. These algorithms include exhaustive approaches, divide and conquer, dynamic programming, greedy algorithms, backtracking, branch and bound, and iterative improvement. We will also cover algorithm analysis, asymptotic notation, recursion, and computational complexity. Data structures covered include vectors, strings, stacks, queues, trees and graphs.

TEXTBOOKS

The course textbooks are:

Introduction to the Design and Analysis of Algorithms by Levitin, 2nd Ed, Addison Wesley, 2007.

Data Structures With C++ Using STL by Ford and Topp, 2nd Ed, Prentice Hall, 2002.

Problem Solving, Abstraction, and Design Using C++ by Friedman and Koffman, 5th Ed, Addison-Wesley, 2006.

PREREQUISITES

Students in EECE3326 may be majoring in electrical engineering, computer engineering, or both. Students come to Northeastern with varying amounts of programming experience in languages like C, C++, and Java, and have a range of programming experiences while on coop. Students taking this class therefore have a wide range of backgrounds and abilities. To take EECE3326 you must:

1. Have successfully completed CS U215, Algorithms and Data Structures for Engineers.
2. Know how to write programs in C++ using arrays, loops, conditional statements, functions, parameters, and simple I/O.
3. Know how to use object-oriented techniques, including classes, constructors, destructors, copy-constructors, public and private members, reference parameters, and function overloading.
4. Know how to use introductory data structures, including arrays, pointers, vectors, stacks and queues, and
5. know how to translate high-level descriptions of algorithms in pseudo-code into working C++.

To review object-oriented programming, read chapters 1–11 and 13 in the book by Friedman and Koffman. To review introductory data structures, read chapters 4–8 in the book by Ford and Topp.

STRUCTURE AND REQUIREMENTS

The course is built around 5 to 7 programming projects (depending on time). Every about two weeks, I will introduce a new project. Lectures will cover the algorithms and data structures needed to solve the problem, and I will answer your questions about the assignment.

Each project will have two parts. In the first part, you will implement the basic infrastructure, i.e., the classes and data structures, for the project. In the second part, you will develop, implement and evaluate algorithmic techniques. There will also be written homework assignments due as part of each project.

Projects can be completed in groups of one or two students. There will be one project submission for each group. However each student must completely understand everything about the solutions they turn in.

Lectures will cover much, but not all, of the required material for the course. There will be reading assignments that accompany each lecture.

There will be a weekly quiz, usually during the first 20/25 minutes of the Monday/Thursday class. Each quiz will evaluate your understanding of the material covered on past projects and lectures. *There will be no midterm or final exam.*

There will be one lab per week in the computer lab (usually on Wednesday. The exact lab classroom will be announced in class and posted on Blackboard). After some lecture, you will spend the remainder of the time in lab working on your projects. The instructor and the TAs will be available to answer questions and give you feedback on your work. In lab, preparation, attendance and participation are required.

GRADING

Your final grade for the class will be based on the following breakdown: 60% quizzes and 40% projects.

Quizzes and projects are graded on an absolute scale (0-100). The class is not graded on a curve. While the lowest of among quiz grades will be dropped, all projects grades will be considered for the final grade.

Labs are graded based on attendance, preparation, and participation. Your lab score cannot improve your grade, but poor performance will reduce your final grade.

Grades are determined from the total points as follows: 96-100: A; 90-95: A-; 85-89: B+; 80-84: B; 75-79: B-; 70-74: C+; 60-69: C; 50-59: C-; < 50: F.

You are strongly encouraged to attend all lectures and labs, complete all projects, and regularly do the reading.

POLICIES

All assignments have fixed due dates and times, by which the assignment files have to be uploaded to Blackboard. No late submission will be accepted (the Blackboard link for the upload will disappear after the due date/time). *Please, do not ask for extensions or exceptions to this simple, fair rule.*

If you miss a quiz, you will not be allowed to retake it. All requirements must be completed during the semester. No incompletes will be given.

Class and lab attendance is required. If you miss a class or a lab, you are responsible for all material that was covered, announcements that were made, and handouts that were distributed in class.

I encourage you to ask questions in class and participate in discussions. However, if I can hear you talking to your neighbor, then your voice is too loud.

If you have a question about the grading of a homework problem you should first contact the teaching assistant directly. If the issue is not resolved to your satisfaction, please contact me.

Changes to project and quiz scores will only be made in the first week after the graded work has been returned to you.

You should check the class website on Blackboard daily for announcements and other information.

You are encouraged to discuss assignments with one another, but all writing of code must be done individually or within your group. You must not look at another student's code, or allow another student to see your code. Copying someone else's work and presenting it as your own, or submitting the same solution as someone else, is not allowed.

You must adhere to Northeastern University's Policy on Academic Integrity. If you violate this policy will receive a lower grade in the course, and may receive an F. You will also be referred to NU's Office of Student Conduct where penalties range from deferred suspension to expulsion from the university.

Exceptions to any course policy may be made if you have a personal emergency that prevents you from participating in the course. In this case you must make arrangements with me as soon as possible, preferably within 24 hours.