

Syllabus

Overview

This course covers the design and implementation of algorithms to solve engineering problems using a high-level programming language. Reviews elementary data structures, such as arrays, stacks, queues, and lists, and introduces more advanced structures, such as trees and graphs and the use of recursion. Covers both the algorithms to manipulate these data structures as well as their use in problem solving. Introduces algorithm complexity analysis and its application to developing efficient algorithms. Emphasizes the importance of software engineering principles.

Instructor

Yifan Sun

Office

R306, 140 the Fenway. You can follow the instruction [here](#) to find my office.

Email

sun.yifa@husky.neu.edu

TA

TBA

Course Schedule

Monday, Wednesday, Thursday 1:35 PM - 2:40 PM @ SH 325

Office Hours

Tuesday, 2:30 PM - 3:30 PM or by appointment

Prerequisites

This course requires either EECE2160 or CS1500. A solid C++ programming skill is required. Students should be very comfortable with writing loops, functions, arrays, pointers, and classes.

Communication

This course uses Piazza as the communication platform. You can put any type of questions and information to Piazza. Private messages are strongly discouraged.

When you ask questions about coding, make sure you follow the [Question Guidelines of StackOverflow](#). Your questions should be general and able to help others that may have the same questions. Also, you have to

state what you have tried to solve your question and why it does not work. Questions like "why my code does not work" will generally not be answered by the instructor.

Collaborations between students are encouraged. Although there will be no extra credit for helping others, you will find teaching and tutoring are very rewarding. If you have general questions such as "why my code does not work", you can still post them on Piazza and I am expecting it can be solved by your peer students.

Assignments

Assignments are given in a mini-project format. There will generally be a mini-project each week. All the mini-projects should be completed individually. All the assignment will allow 2 weeks to solve. All the materials that is required by each homework should be covered in the lectures of the first week. So you should use the first week to complete your homework and treat the second week as extension. No further extension will be granted.

All the assignments are distributed and collected via Github classroom. Therefore, students are required to learn how to use Git by themselves. When an assignment is announced, you will receive a link to accept the assignment. You can follow the link and the instructions to create a repo for your assignment. Your last commit before the deadline will be considered as your final submission.

All the assignments code need to be able to compile and run on a Linux machine. For each assignment, a auto-checking program is provided and when you push to Github, the system will check if you have solved all the coding problems. For any coding problem, if it cannot compile, no more than 20% of the score will be given, and if it cannot pass the auto checking system, no more than 50% of the score will be given. Also, consistent indentation is a very basic professionalism of a programmer. In our assignment, we roughly follow the Google coding guideline. Therefore, please make sure you are familiar with the [Google C++ coding styles](#). If your submitted code is terribly indented, 50% of your score will be deducted. So, after submission, please make sure your code looks well-aligned on Github.

Reference

Text Book

- Thomas H. Corme, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, MIT Press, 2009, ISBN:978-0262033848

Other References

- Stanley B. Lippman, Josée Lajoie, Barbara E. Moo, "C++ Primer", 5th Edition, Addison-Wesley Professional, 2012, ISBN: 978-0321714114

Grading

The grading weight of each is listed in the following table.

Item	Weight
Homework	50%

Midterm Exam	15%
Final Exam	15%
Final Project	20%

Your final grade is calculated as a numeric grade between 0 and 100 based on the percentages shown above, and then converted into a letter grade using the following scale:

High	Low	Grade
100.00	95.00	A
94.99	90.00	A-
89.99	86.66	B+
86.65	83.33	B
83.32	80.00	B-
79.99	76.66	C+
76.65	73.33	C
73.32	70.00	C-
69.99	66.66	D+
66.65	63.33	D
63.32	60.00	D-
59.99	0.00	F

Topics to be Covered

- Unit 1: Foundation
 - Introduction to algorithm and data structure
 - C++ review
 - Algorithm asymptotic complexity analysis
 - Brute force algorithm
 - Divide and conquer
 - Array, dynamically growing array and linked list.
 - Stack, queue
- Unit 2: Sorting
 - Insertion sort and bubble sort
 - Heap
 - Heap-sort
 - Priority queue

- Quick sort
- Unit 3: Containers
 - Hash Table
 - Trees
 - Binary Search Trees
 - Red-Black Tree
 - Trie
- Unit 4: Algorithm Design
 - Dynamic Programming
 - Greedy Algorithm
- Unit 5: Graph
 - Graph representation
 - Breath-first search
 - Depth-first search
 - Minimum Spanning Tree
 - Bellman-Ford Algorithm
 - Dijkstra Algorithm
 - Floyd-Warshall Algorithm

Accommodations for Disabilities

Northeastern University and the Disability Resource Center (DRC) are committed to providing disability services that enable students who qualify to participate fully in the activities of the university. Students with documented disabilities who may need accommodations, or any student considering obtaining documentation should visit the [DRC website](#) or contact their staff at 617.373.2675.

Statement on Academic Integrity

A commitment to the principles of academic integrity is essential to the mission of Northeastern University. The Academic Integrity Policy can be found in the undergraduate student handbook (pages 38-41), or from the [Office of Student Conduct & Conflict Resolution \(OSCCR\)](#). I encourage you to familiarize yourself with it. If a student violates this policy in any way, I reserve the right to impose a sanction of failure on the assignment/assessment or failure in the course. If you have questions about appropriate citations, please ask."