LMU CMSI 281 - Data Structures

Professor Andrew Forney • Andrew.Forney[at]lmu.edu • Fall 2017 • PER 202 • 11:30 am - 12:45 pm

CMSI 281 is a 3-Unit intermediate-level programming class that teaches students how to craft efficient and space-conscientious applications by analyzing and then choosing the "right" data organization for a task. Although Java is the course's language of focus, students will learn core concepts that permeate all programming languages.

Learning Outcomes

By the course's end, students will:

- understand analytic programming concepts such as asymptotics, time-complexity, and space-complexity -- that not all programs are created equal; two programs that perform the same task may not do so with the same efficiency or cleanliness.
- be introduced to a wide range of Abstract Data Types (ADTs) and learn which are best suited for which tasks, including: arrays, linked lists, stacks, queues, heaps, hash tables, sets, trees, and graphs.
- gain a deeper understanding of class hierarchies, classes vs. objects, and implementing ADTs in concrete classes.
- learn the interfaces for popular Java collections, and how to use them in a variety of non-trivial programming tasks.
- identify beneficial and detrimental interactions between ADTs and algorithms that operate on them.

Prerequisites

Warning: before taking this course, it is assumed that you have a sufficient knowledge of the basics of programming (CMSI 185) and Java (CMSI 186).

This class uses Java for its concrete examples, homework, and exams. We will not spend much class time reviewing Java fundamentals, though you are welcome to ask for clarification at any point.

If you require additional practice, you may learn through any number of online tutorials, and test your skills oncodingbat.com (for instance).

Texts

We will be using one textbook for the course, which will primarily cover the theoretical material: Introduction to Algorithms by Cormen, et al.. Although this specific text is not mandatory for the course, it contains all concepts that we will be covering in great detail, and your exams will expect that you understand these concepts.

This is a popular textbook with many editions; any edition will likely match what we cover in the course, so you may purchase whichever you like (if any).

Additionally, I will provide **abbreviated** course notes for our lectures on my course page (see Notes tab)

Resources will be provided throughout the course detailing any additional topics that are not sufficiently contained in the above text. These resources will be free and publicly available.

Lecture Attendance

Although not mandatory, lecture attendance is strongly advised since you'll otherwise miss out on all of our inside-jokes. Less importantly, you'll be missing out on lecture content that will not be posted online, classwork (see below), and hints for homework and exams. If you miss a lecture, the important points will be sketched in our course notes, though may be missing lecture-exclusive information (see Note tab on course page).

Lectures are participatory in nature, often with exercises and group-work to facilitate learning. Laptops are discouraged for note-taking, but encouraged for working on in-class problems.

Workload Expectations

As this is a 3-Unit class, it is expected that you will be allocating an average of 6 hours per week working on course assignments, reading, and studying. Some weeks will have less material to keep you busy, and others more, but be aware that this is a work-intensive course that may require you to spend a lot of time programming!

Piazza Forum

I will be maintaining a Piazza forum for your course questions, located here. You are responsible for reading all homework and exam-clarifying questions that are posted on the forum, as I will also be using this as the chief medium for communicating with the class.

Office Hours

Although you're always free to email me or (preferrably, when appropriate) use the Piazza forum, I'll host some in-person office hours at the following times:

- TR 4:30 - 6:30pm, DOO 201A
- W 2:00 - 4:00pm, DOO 201A

On other days, you'll find me in the Keck Lab or my office (DOO 201A). If my door's open, stop by for any homework questions, lecture clarifications, or chat!

Tentative Schedule of Topics

The following constitutes the tentative schedule for topics to be covered in the course. This is largely an approximation for what we will cover each lecture, and is subject to change.

Item	Date	Topic	Text Pages
Lecture 1T	8 / 29	Introduction, Java - Basics	(course notes)
Lecture 1R	8 / 31	Java - Classes & Interfaces	(course notes)
Lecture 2T	9 / 5	Java - Practice	(course notes)
Lecture 2R	9 / 7	Memory Organization & Contiguous Lists	(course notes)
Lecture 3T	9 / 12	Memory Organization & Contiguous Lists	(course notes)
Lecture 3R	9 / 14	Linked Lists	11.2
Lecture 4T	9 / 19	Linked Lists	11.2
Lecture 4R	9 / 21	Stacks & Queues	11.1
Lecture 5T	9 / 26	Comparative Sorting	(course notes)
Lecture 5R	9 / 28	Comparative Sorting & Review for Exam I	(see above)
Exam I	10 / 3	Concepts Assessed: Lectures 1T - 5T	(see above)
Lecture 6R	10 / 5	Asymptotics & Notation	4.1
Lecture 7T	10 / 10	Asymptotics & Notation	4.1
Lecture 7R	10 / 12	Trees	5.5
Lecture	10 / 17	Tree Traversal & Operations	5.5

8T			
Lecture 8R	10 / 19	Binary Search Trees	13
Lecture 9T	10 / 24	Binary Search Trees	13
Lecture 9R	10 / 26	Heaps	7.1 - 7.3
Lecture 10T	10 / 31	Heap-, Merge-, & Quick-sort	7.4, 8, (course notes)
Lecture 10R	11 / 2	Heap-, Merge-, & Quick-sort	7.4, 8, (course notes)
Lecture 11T	11 / 7	Review for Exam II	(see above)
Exam II	11 / 9	Concepts Assessed: Lectures 5R - 10T (Major Emphasis), Lectures 1T - 5T (Minor Emphasis)	(see above)
Lecture 12T	11 / 14	Analyzing Recurrence	4.3
Lecture 12R	11 / 16	Hash Tables	12
Lecture 13T	11 / 21	Hash Tables	12
Holiday!	11 / 23	No class, Happy Thanksgiving!	N/A
Lecture 14T	11 / 28	Graphs	5.4
Lecture 14R	12 / 1	Graph Operations & Applications	5.4
Lecture 15T	12 / 5	Programming Lab	(course notes)
Lecture 15R	12 / 7	Programming Lab & Massive Review	(see above)
Final	12 /		(see

Exam	1211:00 am	Concepts Assessed: ALL	above)

Assignments & Grading

Grade Decomposition

Grades will be assigned based on the following weighted coursework:

- **[20%] Classwork:** Small classwork assignments will be given throughout each week, with in-class time allocated for their completion. Though you will have the opportunity to finish these in groups and with my assistance during class, you are not required to attend lectures, and so may submit these electronically by their listed due dates. *Your lowest classwork grade will be dropped (even if you skipped it entirely).*
- **[40%] Homework:** Larger assignments with heavy coding required. Programming style and comments are graded as well. All homework assignments are weighted equally. Expect assignments once every ~2.5 weeks.
- **[40%] Exams:** There will be 3 exams every ~5 weeks throughout the semester (including the final). Your best 2 exam scores will constitute 80% of the exam grade, with your worst score constituting the remaining 20%. For example, if your 3 exam scores were 70, 80, and 90, then your weighted exam score for the course will be 82 (= 70*0.2 + 80*0.4 + 90*0.4)

Final Grades

Final letter grades are given based on the university scale of grade percentages:

- A: 93 - 100
- A-: 90 - 92
- B+: 86 - 89
- B: 83 - 85
- B-: 80 - 82
- C+: 76 - 79
- C: 73 - 75
- C-: 70 - 72
- D: 65 - 69
- F: 64 and below

Fractional grade percentages at 0.5 or over will be rounded up, so an 89.5% will be considered a 90% (A-), but an 89.4% will be considered an 89% (B+) on the above scale.

That said, these are only the guaranteed grade assignments: if your "final" grade is an 82%, you are guaranteed a B- or better, but you might still get an A if 82% is the top score.

Extra Credit

Extra credit opportunities will be sporadically available during lectures / exams; if you are present and can hand in an attempt at the extra credit opportunities, you will receive bonus points!

If you receive all extra credit opportunities, you can gain a maximum +2% on your final grade. These bonuses are applied post-curve, so no student will be punished for not attending lectures relative to their peers.

Submission Standards

Each of your submitted homeworks and projects are subject to the following constraints:

- **Assignment posting:** all assignments (classwork, homework, or otherwise) will be announced in class and posted on this site with due dates. You are responsible for checking this site's Announcements for any updates to assignments and their associated deadlines.
- **Late Policy:** assignments are due at exactly the time indicated by the method specified. Assignments that are late by anywhere from 0 - 24 hours will only receive 80% of the points that they would have otherwise earned. Assignments that are late by 24+ hours receive a 0. There are no exceptions to this rule.
- **Individual Work:** students are encouraged to talk and think about problems in groups, but each submission should be their own, containing no code that has been copy-pasted from another student. We take plagiarism very seriously, and each submission will be run through a similarity-checking mechanism to ensure fairness. Code or homework which we believe to be shared innapropriately between students is subject to severe disciplinary action. (not that you would, just sayin')
- **Style:** assignments which are sloppily submitted without proper formatting, style, and comments are subject to penalties. We will discuss what constitutes such offenses in class, though you should be familiar with clean coding standards from your introductory courses.
- **Development Environment:** I will be using the Eclipse IDE in class for demonstration; for your work, you are free to use whatever development environment you wish.

Resources

Tips for Success in this Class

Here are a few tips for succeeding in this class (and college, in general); they're not meant to sound patronizing, but rather, are simply pearls of wisdom to hand down from someone who has recently trodden your path!

- **Ask questions:** if there is one piece of advice I can impart unto you, it's to ask questions when you have them. Chances are good that if you have a question, someone else has the same one, you won't be left behind wondering about something that was covered minutes ago (and which future material might employ), and I promise not to judge you for whatever you ask (well, not too hard at least). At the very least, if you have a question, write it down and then ask it on Piazza after class (which can be done anonymously if you so please). The heart of academic inquiry is just that: inquiry!

- **Fear not the specter of error:** in college classes, there tends to be a crippling aversion to being wrong, especially publicly! I'm here to tell you that there is no better time to be wrong than in class, particularly in one as interactive as ours. There are a variety of silver-linings to being wrong: you won't forget what you were wrong about (one fewer thing to cram for later), it's better to learn from being wrong when it doesn't matter (i.e., in discussion) than when it does (i.e., on a test or during an interview), and I will respect you for your courage. There is only one circumstance in which you should fear error: when you fail to learn from it (at which point, see tip above for remedy).
- **Indulge your curiosity:** and no, that's not some tagline stolen from a Vegas day-spa (actually, it might be, don't quote me on that), but the wisdom is sound... in class, we will cover a variety of topics with accompanying exercises that highlight the main take-away messages of the lesson, but that does not mean that your learning should stop there. Indeed, computer science is one of those beautiful disciplines wherein experimentation and exploration are so trivially entertained. Curious about what would happen if you changed the value of a variable? Do it! Curious about what other methods exist for certain data types? Google them! There is no reason why your learning should be constrained to what I (strategically) offer you; we have only limited time in class, and I'll prioritize the most important topics, but your interests can spread their own wings. Go forth and explore!