CS 2420: Introduction to Algorithms and Data Structures

Course Details and Objectives
Fall 2018

Course Information

Description of CS2420

This course provides an introduction to tools found throughout computer science - basic algorithms and data structures that lend themselves naturally to computational problem solving, as well as the problem of engineering computational efficiency into programs. Students will gain an understanding of classical algorithms (including sorting, searching, tree and graph traversal), and data structures (including linked-lists, trees, graphs, hash tables, and heaps), and an introduction to parallel computing. Students will complete extensive programming assignments that require the implementation, testing, and analysis of these concepts. Students will also gain experience in team programming and software engineering. The prerequisite for the class is CS 1410 (Introduction to Object Oriented Programming). Students are expected to be familiar with the Java programming language.

Meetings

Lectures are Mondays and Wednesdays from 3:00p - 4:20p in ASB 220. Lab sections are Fridays at various times (see course catalog). Please attend the lab section you are enrolled in.

Instructor

Daniel Kopta, *Email:* `dkopta@cs.utah.edu` *Office:* `MEB 3142`

TAs

- Haisley Brooking
- Derek Castillo
- Abhi Dubal
- Bruce Englert
- Jin He
- Josh Homer
- Ram Seethamraju
- William Tang
- Jack Osterloh

Course Materials

Class Website

The class web site uses *Canvas* and is located at:

`utah.instructure.com`

or through the University of Utah web page. It will contain all pertinent course info and materials, such as lecture slides, code demos, and class announcements. It will also contain lab assignments and programming assignments. All of your grades will be posted on this site so you can keep up with them throughout the semester. It is absolutely necessary that you use this web page for this class, and you are accountable for checking it regularly.

Lecture notes

Lecture slides will be posted online shortly before or after each class period, along with any other documents or code examples used in class. This does not mean that all of the material covered for that lecture is included online. Attend class and pay attention!

Lab sections

When enrolling in this class, you also enrolled in a lab/discussion section. These are smaller, weekly meetings led by the TAs designed to give you practice with the concepts discussed in class and other necessary instruction. Lab activities and instructions will be posted online and you must attend and complete discussions for credit.

Grading

Your grade for this course will be determined by the following:

assignments	50%
mid-term exams (2)	25%
final exam	15%
labs and participation	10%

If X is your overall course score, letter grades will be assigned using the below scale. Scores will *not* be rounded.

	$90 > X \geq 87$ B+	$80 > X \geq 77$ C+	$70 > X \geq 67$ D+	
$100 \geq X \geq 93$ A	$87 > X \geq 83$ B	$77 > X \geq 73$ C	$67 > X \geq 63$ D	$60 > X \geq 0$ E
$93 > X \geq 90$ A-	$83 > X \geq 80$ B-	$73 > X \geq 70$ C-	$63 > X \geq 60$ D-	

Assignments

Assignments consist of weekly programming assignments in Java, and will be done in pairs of two, with the exception of the first assignment. The specifications and due dates of these assignments will be posted online each week. You will hand in your programs through the class

web page. Working in pairs is a requirement, and any pair assignment done alone will receive an automatic 15% reduction. Students are required to switch partners at least once. The final assignment will be treated as though you worked alone if you have not switched partners. Partial credit may be given for incorrect programs, but it must be clear that a strong attempt was made. If your program does not compile or run, no credit will be given. Programs will be graded on readability, comments, and design of the code, as well as correctness in execution. Assignments will also consist of a written portion describing the design of the program, and analyzing the computational performance of your solution.

The last assignment of the semester will be a comprehensive project lasting two weeks, in which you will design a solution to a problem from the ground up, with little or no framework given to you. You will have the freedom of selecting which algorithms and data structures are best suited to solving the problem, and designing a complete program around your solution.

Exams

Midterm exams will be given during the regular class time in the regular class room. The final exam will be held on Wednesday, December 12 at 3:30pm, location TBD. All exams are written exams.

Participation

The class participation grade will be based on attendance and completion of lab exercises and in-class responses using TurningPoint. See the class web page to set up your TurningPoint account.

Getting Help

Office Hours

My office hours in MEB 3124 are Mondays and Wednesdays from 11:00 - 12:00. You are welcome to stop by any other time if I am there. In general, it is best to come during my office hours or email me to set an appointment to guarantee that I will be there.

TA Consulting Hours

The teaching assistants will have regularly scheduled consulting hours held in the CADE lab (WEB L224/226). They will help with any course content, and are primarily there to help you with the weekly homework assignments. See the course web page for the TA consulting schedule.

Other Issues

Send email to the teach staff via Piazza if you have an issue or questions that can't be resolved on the student help forums, such as a grading issue, or if your question requires posting code. See the "Getting Help" page on Canvas to set up your Piazza account. If you have a question about a personal matter, or one that does not relate to the course, feel free to email me directly at `dkopta@cs.utah.edu`.

We will send email and announcements to everyone in the class, such as corrections to assignments, changes to due dates, clarifications, etc. through the class web page. Students are required to check their email and the class web page regularly.

Course Guidelines

Working Together

Most of the work in this class will be completed in pairs with another student. When pair work is required, students must adhere to the techniques of pair programming. Partners are required to contribute equally half of the work and problem solving required. Students are encouraged to discuss high-level solution strategies with fellow classmates, but each student is responsible for writing her/his own answer. Cheating is: sharing (outside of a partnership) written or electronic work either by copying, retyping, looking at, or supplying a copy. Cheating is not: discussing concepts, answering questions about concepts or clarifying ambiguities, or helping someone understand how to use the class tools and software. There must be no collaboration during exams. See the class web page for the cheating policy for this course. See the University of Utah Student Code for a detailed description of the University policy on cheating. Any student found cheating will fail the course. Supplying cheated materials is considered cheating just as using them is.

College of Engineering Guidelines

For information on withdrawing from courses, appealing grades, and more, see:

`coe.utah.edu/students/academic-affairs/academics/semester-guidelines/`

Attendance

By enrolling in this course you are implying your availability to complete and attend all lectures, labs, assignments, and tests. There will be no extensions on assignments or makeup tests due to absence, unless there is a legitimate emergency or civic duty. Specifically, vacation time and job responsibilities are not a legitimate emergency.

Late Work

Students are allotted five late days to use throughout the semester on assignments. These are to cover emergencies and special circumstances beyond your control. Do not use your late days for non-emergencies. After using your late days, further late work will receive no credit. Late days apply to programming assignments only; other work, such as labs, will not be accepted late.

Students with Disabilities

The University of Utah seeks to provide equal access to its programs, services, and activities for people with disabilities. If you need accommodations in this class, reasonable prior notice needs to be given to the Center for Disability Services, 162 Olpin Union Building, 581-5020 (V/TDD). CDS will work with you and the instructor to make arrangements for accommodations.

Course Objectives

- Students will improve on the basic object-oriented programming skills learned in CS 1410. Understanding of the concepts of inheritance, polymorphism, and generic programming will be strengthened significantly by creating solutions with multiple levels of inheritance and by implementing generic versions of data structures.
- Students will learn algorithms for searching (sequential and binary) and sorting (selection, insertion, Shellsort, mergesort, quicksort, heapsort), as well as the asymptotic behavior of each.
- Students will gain more experience using and implementing recursion. Further, students will receive preliminary instruction as to when to apply iteration instead of recursion.
- Students will become proficient using (and in most cases, implementing) data structures fundamental to computer science including: arrays, linked lists, stacks, queues, graphs, trees, binary search trees, hash tables, binary heaps, and priority queues. Further, students will reason about the Big-Oh behavior of basic operations on such data structures.
- Students will regularly evaluate their solutions for efficiency (often in writing). This may include efficiency in time (Big-Oh and/or actual running time), efficiency in space (memory requirements), and/or programmer efficiency (ease of implementation and maintenance).
- Students will methodically test their solutions according to a variety of testing models discussed in class.
- Students will gain experience designing, implementing, and testing solutions in pairs using the techniques of pair programming.