

CS 2420: Intro to Algorithms and Data Structures

Administrative Details and Syllabus Spring 2020

Important Information

Class Website	Canvas (available through CIS)
Lectures	Tuesdays and Thursdays 9:10-10:30a in 2230 WEB
Labs (attendance required)	Fridays 9:40-10:30a, 10:45-11:35a, or 11:50a-12:40p in L130 WEB
Instructor	D. Erin Parker, 3144 MEB
Textbook	none (reading assigned from free, online sources)
Exam Dates	<u>Mark your calendar – exams may not be missed!</u>
Midterm exam 1	Thursday, February 20 (during class)
Midterm exam 2	Thursday, April 2 (during class)
Final exam	Monday, April 27 3:30-5:30p
Final course grade	Programming assignments and analysis documents 45%, Exams 40%, Labs 10%, Online quizzes 5% <i>Failing exam average</i> → <i>failing course grade</i>
Getting help	The class website has details on how to see TAs and the instructor outside of class, as well as how to post questions to the class forum and contact the course staff.

Course Description

This course provides an introduction to the problem of engineering computational efficiency into programs. Students will learn about classical algorithms (including sorting, searching, and graph traversal), data structures (including stacks, queues, linked lists, trees, hash tables, and graphs), and analysis of program space and time requirements. Students will complete extensive programming exercises that require the application of elementary techniques from software engineering.

The prerequisite for this course is CS 1410: Introduction to Object Oriented Programming.

Course Materials

Website. The class website is a Canvas course available through CIS. *It is always under development*, with updates to the class schedule, lecture notes, assignment specifications, and more, occurring regularly. It is critical that students become familiar with the class website right away and *plan to visit it several times a week, at a minimum.*

Textbook. Reading is assigned from online notes and articles linked from the class website. There is no required textbook.

Pencil/pen and paper. Students should bring a pencil/pen and paper to every lecture and lab session. In-class problem solving, in particular writing source code by hand, is used regularly to prepare students for exams. Laptops, tablets, phones, and other devices are not permitted during exams; likewise, they may not be used for in-class problem solving exercises.

Lecture notes. The instructor often makes use of typed notes, sample source code, and other materials during lecture. These items are posted on the class website following the lecture; however, such posted items may not represent completely the material covered in class. Students who must miss class are strongly encouraged to check with a classmate.

Laboratory practice. Lab sections meet on Fridays at the times listed. Students must attend the lab section for which they are registered, unless prior arrangements are made with the instructor. Students should bring their own laptops or plan to use a “loaner” laptop for the 50-minute lab session.

Personal computers. Students may use their own computers for completing assignments and online quizzes; however, broken tools or computers, or network connectivity issues are not sufficient basis for a deadline extension. Plan ahead and use the CADE lab machines if your own computer is not working.

Student Evaluation

Programming assignments and analysis documents. The instructions for each assignment and its due date are posted on the class website at least one week before it must be submitted. It is the student’s responsibility to ensure the successful and timely submission of each programming assignment — start early and follow the instructions carefully. Corrupted or missing files are not grounds for extensions — double-check your submissions and save a digital copy of all of your work in your CADE account. *The timestamps of files outside of your CADE account cannot and will not be trusted.*

Late submissions are penalized 10 points for each day beyond the deadline, for a maximum of three days. Note that 12a marks the start of a new day and -10 points.

Lab exercises. Every Friday, students complete a lab exercise reviewing the material covered recently in lecture or preparing for an upcoming assignment. Often, a worksheet accompanies the lab exercise and is due at the end of the designated lab period. *No lab exercise / worksheet may be made up for credit, except in the case of a documented medical emergency.*

Quizzes. Students take online quizzes regularly, reviewing the material covered recently in lecture or preparing for an upcoming assignment. *No quiz submissions will be accepted late, except in the case of a documented medical emergency.*

Exams. Midterm exams will be given during class meetings on February 20 and April 2. The final exam is cumulative and will take place 3:30-5:30p April 27. (Note that this is scheduled as a departmental exam and does not follow the University’s “Day Class Exams” schedule.) *No exam may be taken at a different time for any reason other than a documented medical emergency.*

Final course grade. *If the average score for Midterm Exam 1, Midterm Exam 2, and Final exam is 65% or lower, the final course grade will be no higher than a D+. Otherwise, the final course grade is based on programming assignments and analysis documents (45%), exams (40%), labs (10%), and quizzes (5%).*

Regrades. Students who wish to appeal a score on an assignment, a lab, a quiz, or an exam must do so **within one week of receiving the score**. For an assignment or exam, use the Regrade Request system in Gradescope. For a Canvas quiz or lab, post a private message to Piazza in the “quiz-regrade” or “lab-regrade” folder.

Letter grades. The following table is used to associate numerical scores with the corresponding letter grade, *notice that scores are not rounded*.

$93 \leq X \leq 100$	A	$87 \leq X < 90$	B+	$77 \leq X < 80$	C+	$67 \leq X < 70$	D+	
$90 \leq X < 93$	A-	$83 \leq X < 87$	B	$73 \leq X < 77$	C	$63 \leq X < 67$	D	$X < 60$
		$80 \leq X < 83$	B-	$70 \leq X < 73$	C-	$60 \leq X < 63$	D-	

Getting Help

To get help understanding course material, students may see the Teaching Assistants during *TA Consulting Hours*, see the instructor during *Office Hours*, post a question to the Q&A forums on Piazza (<https://piazza.com>), or contact the course staff directly (also via Piazza). See the **How to get help in CS 2420** module on the class website for details.

Behavior in the Classroom

Student code. All students are expected to maintain professional behavior, according to the University of Utah Student Code at www.regulations.utah.edu/academics/guides/students/studentRights.html. Students should read the Code carefully and know that they are responsible for the content. According to Faculty Rules and Regulations, it is the faculty responsibility to enforce responsible classroom behaviors, beginning with verbal warnings and progressing to dismissal from class and a failing grade. Students have the right to appeal such action to the Student Behavior Committee.

Laptop policy. Students are expected to engage with the instructor and classmates during lecture. Students are permitted to use a laptop or mobile device to take notes or respond to in-class polling. Use of a laptop or mobile device for any other purpose is not permitted, and students who do so will be asked to leave the classroom.

Other guidelines. See also the School of Computing Policies and Guidelines, the College of Engineering Guidelines, the Guidelines and Policies for Pair Programming, and the CS 2420 Academic Misconduct Policy. All of these documents are linked from the Canvas class website (under Syllabus). *Read them and ask questions, as needed – you are responsible for the content.*

Course Objectives

The following are the expected outcomes for a student completing CS 2420.

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