

# CS 146 (Sections 4, 5): Data Structures and Algorithms, Spring 2023

## David Scot Taylor

Associate Professor

[Dept. of Computer Science](#)

[San Jose State University](#)

212 MacQuarrie Hall

Phone: (408) 924-5124 (email works better)

Email: david.taylor "at" sjsu.edu

Spring 2023 office hours:

- Monday, 1:30-2:30 (in person).
- Tuesday, 12:15-1:15 (in person).
- Other times available, especially on Tuesdays. Set up an appointment by email.

Office hours will generally be in-person. When they are virtual, use [this zoom link](#) for zoom office hours, and you will enter the waiting room (I will see one at a time in the order you enter, occasionally updating the waiting room through chat about the size of the queue.)

## Class Meetings:

- Section 4: Mon/Wed 9:00-10:15am, MacQuarrie Hall 223.
- Section 5: Mon/Wed 10:30-11:45am, MacQuarrie Hall 223.

## Prerequisite Courses

You must show me that your prerequisite courses have been satisfied. **If you have not submitted prerequisites by the start of Monday, February 13, you might be dropped from the course, if other students are waiting for your space.** (Drop date is one week later, but that is also add date and other students need to know if they can get into this class in case they need to drop a different class. I will not give out any add codes without first seeing prerequisite proof.)

The prerequisite courses are: Math 30, Math 42, and (CS46B in Java or (CS49J and CS46B)). You should show me grades for CS46B, Math30, and Math42, or their equivalents on a San Jose departmental course equivalence form. You must have a C- or better in each course. If you took CS46B elsewhere, you should also show me either proof that you have taken CS49J, or proof that your CS46B was in Java.

Prerequisite courses and what they covered:

CS46A: Introduction to programming, conditionals, iteration, and recursion.

CS46B: Stacks and queues, lists, dynamic arrays, binary search trees. Iteration over collections. Hashing. Searching, elementary sorting. Big-O notation. Standard collection classes.

CS49J: Java programming language, if your CS46A/B were in a different language.

Math30: Ideally, you would have sequences and series (Math 31), but proof of Math 30 will suffice.

Math42: Sets, logic, proofs, induction, combinatorics, probability, and equivalence classes.

## Course Format

I have already created videos to introduce the algorithms, data structures, and analysis of this class. Because of that, we will use class time to introduce ideas through problem solving, work on difficult problems, and in a few cases, code review. Reading, videos, rote homework problems, and programming will generally take place outside of class time.

## Course Website

The main course website will through SJSU's [Canvas](https://sjsu.instructure.com/) website at <https://sjsu.instructure.com/>. Basic course information, including a link to this greensheet, is at <http://www.cs.sjsu.edu/faculty/~taylor/term/current/CS146/>.

## Course Description

Implementations of advanced tree structures, priority queues, heaps, directed and undirected graphs. Advanced searching and sorting (radix sort, heapsort, mergesort, and quicksort). Design and analysis of data structures and algorithms. Divide-and-conquer, greedy, and dynamic programming algorithm design techniques.

## Course Goal

To examine various ways to represent data used by programs and to compare these representations in terms of their memory requirements and the resulting program execution times.

Time will be spent learning algorithms and data structures, mathematical tools and techniques (recursion, recurrence relations) useful for their design and analysis, and seeing some examples of when they are needed.

## Course Objectives

- To ensure that students are familiar with ways to implement elementary data structures and their associated algorithms.
- To introduce students to the implementation of more complex data structures and their associated algorithms.
- To acquaint students with advanced sorting techniques.
- To teach students how to determine the time complexity of algorithms.
- To introduce students to algorithm design techniques.

## Course Learning Outcomes

Upon successful completion of this course, students should be able to:

- Understand the implementation of lists, stacks, queues, search trees, heaps, union-find ADT, and graphs and be able to use these data structures in programs they design
- Prove basic properties of trees and graphs
- Perform breadth-first search and depth-first search on directed as well as undirected graphs
- Use advanced sorting techniques (heapsort, mergesort, quicksort)
- Determine the running time of an algorithm in terms of asymptotic notation
- Solve recurrence relations representing the running time of an algorithm designed using a divide-and-conquer strategy
- Understand the basic concept of NP-completeness and realize that they may not be able to efficiently solve all problems they encounter in their careers
- Understand algorithms designed using greedy, divide-and-conquer, and dynamic programming techniques

# Required Texts/Readings

## Textbooks

This textbook is very widely used, and I hope it will come in handy beyond this course. The 3rd edition, for the material we cover, is quite similar to the 2nd edition. (The 2nd edition managed to obfuscate a few issues from the 1st edition while clarifying others.) I think the majority of changes from the 2nd to the 3rd edition are in sections we don't cover, though some of the exercises and readings have changed. When possible, I will post assignments for both the 2nd and 3rd editions of the book.

Introduction to Algorithms, 3rd Edition  
Cormen, Leiserson, Rivest, and Stein  
ISBN-10: 0262033844  
ISBN-13: 978-0262033848  
MIT Press, 2009

You can find errata (bug reports) for the book <http://www.cs.dartmouth.edu/~thc/clrs-bugs/bugs-3e.php>, for whichever printing of the book you get.

## Other Technology Requirements

Students are expected to have wireless laptops, with Java and a Java IDE installed.

## Other Readings

I will make any additional reading material as needed, either by a link or by hardcopy.

# Course Requirements and Assignments

## Workload

The following will be regularly assigned for time outside of class:

- Video lectures
- Rote homework problems or video quizzes given in Canvas.
- Readings from textbook or handouts
- Canvas Quizzes covering other material (this has replaced written exercises of the past, but perhaps sometimes there will still be written questions for submission)
- 4 Programming assignments
- 2 Practice Exams

During the introduction of new material, homework is our chance to learn by making mistakes. It is expected that you will make an effort in all of the above for the sake of learning, and to give yourself feedback about your understanding the material.

The purpose of the rote homework is to give you enough practice working on problems to either understand how to solve the problems, or at least to learn from solutions to those problems.

I encourage you to watch each video closely and carefully, pausing and taking notes if needed, before attempting homework on that video. If you don't do well on your first attempt at the homework, it indicates that you have not watched the video closely enough. This should give you some immediate feedback on whether or not you are watching the videos closely enough.

Canvas quizzes on the videos can (usually) be taken up to 5 times, but it is really your first attempt that will give you the most feedback on how well you understand the material. Using what would be a terrible homework assignment to give an example of the difference between the first attempt and later attempts: imagine that you are told to study the first 15 digits of pi. After doing that, if you were asked "What digit is in the 10 millionths position?" and you could answer that, it would give some indication that you had memorized the requested digits. On the other hand, if you got it wrong, and then looked at the digits of pi before taking a second attempt to answer the question correctly after memorizing the single digit requested, it would give little confidence that you knew any other digits of pi. Hopefully, your homework will be much more interesting than that example.

If, even **after a second attempt** at a Canvas homework, you still cannot get the correct answers, you can ask a classmate or somebody else to help you. **You may not simply ask them for the answers. (That counts as academic dishonesty.)** You can, on the other hand, have them walk you through the problem, step by step, until you get to an understanding of the answer. (For this course, that counts as diligence. You should understand your submitted answers, otherwise you need to spend more time on them.) Because the homework has a time limit for submission, you might want to take your 3rd attempt, if needed, to record the questions, so that you can do them with more time on your 4th attempt. You really shouldn't need a 5th attempt. The reason attempts are limited is to stop anybody from making blind submissions until they happen to get the correct answers.

In Spring 2021, I shifted away from written homework problems, but if I do have any, they will not be graded on correctness, but on whether or not it looks like an honest effort attempt was made to answer that problem. **You may discuss problems with others, but anything other than superficial comments must be documented.** You should not simply copy solutions, nor look for solutions (on the web or elsewhere), but if needed you can have somebody explain a problem to you in full, until you understand the solution. I might only return solutions for those problems for which you turn in evidence of putting in enough effort.

**Note: The first homework assignments will be available until mid February, while the class enrollment is still being worked out. After that, homework will only be available until it is due. If you submit the homework, you will still be able to see it afterwards, but if you don't get around to looking at your homework before its due date, it won't be.**

For both Canvas and written homework, you should do each homework, unless you are positive that you understand the topic so well that doing the homework would not be a good use of your time. And for those students? They should be the ones helping classmates to understand the material, as outlined in the two preceding paragraphs.

Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.

## Class Participation

Class participation and feedback are very important to keep the course interesting. *If I am covering material too slowly or quickly, or if I am not clearly explaining things, you must let me know.* I prefer an interactive learning environment. If you disagree with something I say, speak up. Argue with me in front of the class. It will make the class better, and right or wrong, constructive interaction will not hurt your grade. If you are correct, clearly my mistake should be corrected. If you are incorrect, probably I have not explained something clearly anyway, and at least half of the class is confused by it. Point it out right then and there. In cases of exceptional participation that seem to benefit the class as a whole, I reserve the right to improve a student's grade by up to 1/3 grade.

The class was designed to be taught in a 'flipped' style: during class time, we will try to spend our time on interactive activities, rather than me lecturing to you. Many students feel like this style doesn't teach them as

well. [Research](#) shows the strong possibility that it is more effective, even if students don't feel that way.

## University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' Syllabus Information web page at <http://www.sjsu.edu/gup/syllabusinfo/>.

## Drop and Add Dates

Note that for this semester, the last day to drop without consequence is **Monday, February 20**. That is also the last day to add. After that date, it becomes very difficult to drop or add a class, so be sure you are where you want to be before these dates arrive! If you are going to drop the class, figure that out ahead of time, so that there is time for you (and others) to shuffle courses as needed.

## Class Format

This class is scheduled to be an in-person class.

## COVID-19 and Monkeypox

Students registered for a College of Science (CoS) class with an in-person component should view the [CoS COVID-19 and Monkeypox Training](#) slides for updated CoS, SJSU, county, state and federal information and guidelines, and more information can be found on the SJSU [Health Advisories](#) website. By working together to follow these safety practices, we can keep our college safer. Failure to follow safety practice(s) outlined in the training, the SJSU Health Advisories website, or instructions from instructors, TAs or CoS Safety Staff may result in dismissal from CoS buildings, facilities or field sites. Updates will be implemented as changes occur (and posted to the same links).

## Grading

### Homework:

Homework is for learning material, and for giving you feedback. Grading homework in the traditional sense encourages students to focus on grades more than learning the material, which is not considered to be the best way to learn. Because of this, for this class, an individual's homework scores will not have a direct effect on their grade as an individual. They will at most have a small, indirect effect, as described below. Of course, proper effort on homework by an individual will likely result in higher exam scores for that individual.

Instead, homework scores will be slightly used to set the curve for the exams: if the class as a whole is doing its homework, it might push the median test score towards the B- range, while if the class, as a whole, is not doing homework, it might push that median towards the C+ range.

I do **not** believe in assigning homework for the sake of assigning homework. If you understand the material without doing the homework, and the homework seems like a waste of time to you? In that case, your advanced understanding should be reflected by your performance on the rote tests (described below), with the expectation that you should do well on those tests. If you are in the top half of the class on those exams, your homework score will be ignored when setting the curve for the class, so your lack of homework will not hurt anybody. You are encouraged to help your classmates to learn the material, teaching it to them will help you to understand it even more.

Note, **this policy should highly discourage students from cheating on homework:** the (admittedly small) risk of being caught is balanced by the fact that your individual homework score will not help you individually, it will only play a very small role in setting the curve for the exams. Because your homework is only one of many homeworks used to set that curve, in cheating, you are taking all of the individual risk, while not reaping any individual reward. Additionally, you won't actually learn the material.

Due to the way that Canvas homework (multiple attempts) and written homework (graded on effort) are graded, the students making a real effort on homework should, as a whole, expect to get credit for a high percentage (90%?) of homework problems. Why should you do the homework? To learn the material. **In some sense, homework is not optional, it is mandatory!** It just doesn't happen to be part of your weighted course grade.

**Rote Tests:** There will be two in-class tests in the final weeks of the semester, that will test "rote" knowledge. You will be given a template for each exam a week in advance. These two exams combined will be scored on a (curved) scale from F to B-. (For the purpose of the class, you will just get a letter grade on the exam.) The curve will be based on both test performance, and my impression of the class performance as a whole (from homework and class interaction). **You are expected to have a laptop computer, with the lockdown browser installed, that you can use for the exams.**

## Final Exam:

- For Section 4, the final exam will be Tuesday, May 23, at 7:15am-9:30am
- For Section 5, the final exam will be Monday, May 22, at 9:45am-noon

Your final exam will be a mixture of rote and advanced questions. I will give you some practice problems (during the semester) to give you some idea of what that means, but you will **not** get a template for the exam. It will be graded on a curve, from F to A+. **You are expected to have a laptop computer, with the lockdown browser installed, that you can use for the final.**

## Programs:

You are expected to work on all programs, to submit them, and to be able to explain them to me. You will have individual code review with me to discuss your code. (For the semester, three regular class meetings will actually be canceled, to make more time for code review, for the first two programs.) Different programs are worth different numbers of points (2, 4, 3, and 2 respectively), for 11 points total. For each program, if you are able to program enough of the work, and explain it to me, you might still get full credit for the program, even if it does not answer every test case. A program that works perfectly that you cannot explain will not get credit, and may violate academic honesty policies. Your ``programming" score starts at -11, and the more of your program you complete, the closer to 0 you get.

## Course grade

First, I will take the maximum of your Rote Test grade, and your Final Exam grade, to get your exam grade. If you have at least 8/11 points from your programs, that will be your course grade. From 5.5/11 up to but not including 8/11 points will deduct 1/3 of a letter grade. From 3/11 up to but not including 5.5/11 will deduct 2/3 of a letter grade. Finally, under 3/11 points will deduct a full letter grade. The resulting modified exam score will be your course grade.

## Program Collaboration Policy

You are expected to code your own programs, with at most minor help from others. Talking abstractly, while still trying to figure it out, with somebody else who is in the same situation, is fine. Sharing code is not, and this includes reading their code and retyping it, or having them dictate it to you. Do not look for premade solutions. **Do not copy code.** You should understand what your code does. If I ask you what something does in your code,

and you don't understand why it is in your code or what it does? That is unacceptable, as it indicates that you are submitting work which is not your own. If you can get somebody to explain something to you in detail, to the point that you can understand and code it, that is okay. **Your code will be checked for correctness, but graded on your ability to answer questions about it. It is possible to get credit for code that doesn't work. It is possible to not get credit for working code if you don't seem to understand what it does. This latter case may also be deemed academic dishonesty.**

Although talking with others who are working on the problem is fine, do not directly tell somebody else how to do it if you have already figured out an algorithmic aspect and have yours working. If I think it is appropriate to give algorithmic hints, I will give them, that is my role as the instructor. They should be asking me, not you. **Beyond scheduled office hours, I will spend over 100 hours this semester answering emails and holding additional office hours. Answering questions, based on the students situation, is a large part of my job as an instructor.** I might not always give as direct an answer as you are hoping for, it depends on how far you have gotten, what you have tried, and what the deadlines are. But, I really do try to give you answers that will push you towards learning the material. Do not get the answers from someone who has already finished the programs.

I run a code plagiarism check at the end of the semester. Having conversations about copied code with students, and writing reports on academic integrity is one of the few parts of this job that I hate. Please don't put yourself, or me, into this position. There will be an assignment on Canvas to state that you understand the homework and program policy, and that you will not share code, nor use someone else's code.

## Recording Lectures or Sharing Course Materials

You can make audio recordings of class for your own personal use. Perhaps you want to want to have my dulcet tones lull you to sleep at night instead of only during class, that is fine. Weird, but fine. Perhaps you want to torture your neighbors by blasting it on your porch, that is not fine: aside from possible violations of the Geneva Convention, recordings should not be reproduced, distributed, or publicly broadcasted. If you want to make video recordings, please discuss it with me.

Course material developed by the instructor is the intellectual property of the instructor and cannot be shared publicly without his/her approval. You may not publicly share or upload instructor generated material for this course such as exam questions, lecture notes, or homework solutions without instructor consent.

## Tentative Class Schedule

A precise schedule will be online in the school's Canvas system. Below is the planned outline, which will be modified as we go.

Date Subject to change	Planned Topic	Actual Topic (if different) or more detail
January 25	Introductions, Administrivia, Warm-Up	
January 30	Finish Warm-Up, Harder ProblemStart Application	
February 1	Defining Problems and Loop Invariants	
February 6	Asymptotic Notation Exercise	
February 8	Heap Lab: code review/lab exercise	<b>No class today:</b> individual code review by appointment (through canvas)

		for everyone either Tuesday February 7, Wednesday February 8, or Thursday, February 9.
February 13	Recurrence Relations, Exercise	
February 15	Master Theorem or Sorting Questions? Exercise	
February 20	Quicksort/select Questions? Exercise	
February 22	Problem Lower Bounds, Exercises	
February 27	Finish exercise from previous class, lower bounds by reduction	
March 1	23-Tree Code Review	<b>No class today:</b> individual code review by appointment for everyone in the class, today or tomorrow
March 6	Linear Sorts, Exercise	
March 8	23-Tree Code Review	<b>No class today:</b> individual code review by appointment for everyone in the class, today or tomorrow
March 13	Graph Exercise	
March 15	Graph Exercises	
March 20	Graph Exercise	
March 22	Graph Exercise	
April 3	Exercise	
April 5	DP Questions?	
April 10	Exercise	
April 12	Rod Cutting, other Dynamic Programming Questions	
April 17	Subset Sum Questions Intro to NP	
April 19	NP	
April 24	NP decision vs. Optimization	
April 26	?	
May 1	Review Practice MT 1 Advanced Topic for Fun	
May 3	Review Practice MT 2 Advanced Topic for Fun	
May 8	Rote exam 1: through B-Trees (approx)	
May 10	Rote exam 2: after B-Trees (approx)	

May 15

Exams returned, Review for  
Final.

- For Section 4, the final exam will be Tuesday, May 23, at 7:15am-9:30am
- For Section 5, the final exam will be Monday, May 22, at 9:45am-noon