

# Data Structures and Algorithms

## CS 146

Summer 2025 Section 01 In Person 3 Unit(s) 06/02/2025 to 08/08/2025 Modified 06/03/2025

### Contact Information

---

Instructor(s): William "Bill" Andreopoulos

Office Location: MacQuarrie Hall 416

Telephone: (408) 924-5085

Email: [william.andreopoulos@sjsu.edu](mailto:william.andreopoulos@sjsu.edu)

Office Hours: Friday 9:00-11:00am (Zoom)

Class Days/Time: Monday and Wednesday, 1:00-3:00pm

Classroom: MQH 225

### Course Information

---

#### Course Format

This course adopts an in-person classroom delivery format.

#### Faculty Web Page and MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on Canvas Learning Management System course login website at <http://sjsu.instructure.com>. You are responsible for regularly checking with the course messaging system to learn of any updates. You should modify the Canvas settings for notifications of announcements and Slack messages to be sent to you.

### Course Description and Requisites

---

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

Prerequisite(s): MATH 30, MATH 42, CS 46B, and [(CS 48 or CS 49J) if CS 46B was not in Java], each with a grade of "C-" or better; Computer Science, Applied and Computational Math, Forensic Science: Digital Evidence, Software Engineering, Data Science majors only; or instructor consent.

Letter Graded

## \* Classroom Protocols

---

### Communication with the instructor

As this is an in-person section, course-related communication should preferably be done in-person during the regular class meeting time (in-person or via Zoom) or after class or office hours. For online communication, students should use the course channel. Rather than emailing redundant questions to the teaching staff, students should post questions on the course channel where the entire class can read and benefit from the responses. The system is catered to getting students help efficiently from classmates, the TA, and the instructor. *Private messages sent to the instructor's other email addresses may get lost due to the large volume of emails received.*

The professor responds primarily to the channel. The professor will re-post questions that are of general interest (e.g. about homework) or discuss them in class. Students are responsible for everything said in class. It is students' responsibility to keep up with what is said in class and not re-post the same questions repeatedly.

When students use the provided online channel, they are expected to be identifiable through their names. Anonymous postings are unacceptable. Students who use fake nicknames may be removed from the channel. If your tone and style of writing is inappropriate the professor may remove you from the channel. The professor should be informed about any other Discord channels used by students. The instructor does not write messages after normal business hours, on weekends or holidays.

Reviewing code for the homework and technical trouble-shooting should be done during the office hours.

Never email your entire code for an assignment to the instructor. The instructor will not fix all the bugs in your code. Limit the code you post to 20 lines or less.

Announcements that concern everyone, such as reminders about due dates or class policy, will be posted.

### Graders/TAs

Sushmitha Chikkanayakanahalli Gopala [sushmitha.chikkanayakanahalligopala@sjsu.edu](mailto:sushmitha.chikkanayakanahalligopala@sjsu.edu)

# Class Attendance

Attendance (in-person or via Zoom) is highly recommended. Classes will be recorded as Zoom screencasts and posted on Canvas. Students are responsible for all material presented in class.

The polling questions in the slides are in the form of multiple-choice and true-false questions. Students should participate and follow the polling questions, either via Zoom polling or Zoom chat or ask in class.

## Regrading Procedure

Grades assigned are final, unless there was an error in the grading. Special requests (e.g. grade changes or deadline extensions) should be done in-person; such special requests sent via electronic messages to the teaching staff will not be honored, since this is an in-person section. To request a higher grade, students should first submit the Canvas "Regrade request" form so there is a record of the request. After submitting a regrade request, please speak with the professor in-person during office hours or after class. Grades may be reevaluated at anytime and may go down as a result of a regrade.

At the end of the semester grade roundups (e.g. 89.95% to 90%) will only be considered if a student has pursued any extra credit opportunities offered and completed the SOTE evaluation.

## Classroom Protocol

Students on Zoom should be muted when not speaking, and must be dressed appropriately when their camera is on.

The class might be made remote if a heat event occurs on a day.

Course material developed by the instructor is the intellectual property of the instructor. Students can not publicly share or upload instructor generated material for this course such as exam questions, lecture notes, hands-on exercises or homework solutions without instructor permission.

## Program Information

---

Diversity Statement - At SJSU, it is important to create a safe learning environment where we can explore, learn, and grow together. We strive to build a diverse, equitable, inclusive culture that values, encourages, and supports students from all backgrounds and experiences.

## Course Learning Outcomes (CLOs)

---

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

1. 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
2. Prove basic properties of trees and graphs
3. Perform breadth-first search and depth-first search on directed as well as undirected graphs
4. Use advanced sorting techniques (heapsort, mergesort, quicksort)

5. Determine the running time of an algorithm in terms of asymptotic notation
6. Solve recurrence relations representing the running time of an algorithm designed using a divide-and-conquer strategy
7. 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
8. Understand algorithms designed using greedy, divide-and-conquer, and dynamic programming techniques

## Course Materials

---

### Textbooks

Cormen, Leiserson, Rivest and Stein, Introduction to Algorithms, 3rd Edition. (*CLRS*)

ISBN-13: 978-0262033848

ISBN-10: 9780262033848

MIT Press, 2009

You can find [errata \(bug reports\)](#) for the book:

<http://www.cs.dartmouth.edu/~thc/clrs-bugs/bugs-3e.php>.

CLRS [e-textbook](#) is available via the SJSU library:

[https://sjsu-primo.hosted.exlibrisgroup.com/permalink/f/1cue0e3/01CAL\\$ALMA51438951350002901](https://sjsu-primo.hosted.exlibrisgroup.com/permalink/f/1cue0e3/01CAL$ALMA51438951350002901)

Sedgewick and Wayne, Algorithms, 4th Edition. (*SW*)

ISBN-13: 978-0321573513

ISBN-10: 032157351X

This book and its [website](#) contain Java implementation of many algorithms covered:

<https://algs4.cs.princeton.edu/code/>

### Other technology requirements / equipment / material

Java Compiler (version 7 or later) and Eclipse.

## Course Requirements and Assignments

---

SJSU classes are designed such that in order to be successful, it is expected that students will spend a minimum of forty-five hours for each unit of credit (normally three hours per unit per week), including preparing for class, participating in course activities, completing assignments, and so on.

**Reading assignments:** Readings will regularly be assigned for the next class (see schedule). Slides will be posted under the Canvas modules before the next class.

**Homework assignments:** Programming assignments will be assigned. More information will be given at the time of the first programming assignment. Late penalty is 5% per day, until 14 days after the original due date (or the last day of classes if it is sooner) when the submission page will close. After the closing date, homework will not be accepted and will receive zero, unless there is a medical excuse with a physician note.

While it is fine to discuss the worksheet/assignment solutions with your peers, code solutions submitted on Canvas should reflect the student's own efforts in writing the code. *Do not write the code for anyone else. Never copy and paste code you find on another source, such as a website, especially if you don't understand the code.*

All homework should be submitted on Canvas. Homework sent via an email or message will not be graded. Homework cannot be graded after it has been reviewed in class or the solution has been posted.

**Quizzes:** There are quizzes posted on Canvas for each module. The quizzes are open-book. The quizzes stay open till the end of the semester. You can retake the quizzes multiple times. The highest grade you achieve is the one that counts for grading.

**Worksheets:** There will be worksheets with problem solving. Worksheets are optional, but everyone will be assigned to present the solution of one worksheet or assignment or quiz to the class (presentations will be assigned in alphabetical order by name). Presenting the code-based homework will involve a code review by walking us through the code and demoing it. Presenting pen-and-pencil-homework can involve using a gDoc or gSlides or the whiteboard. The grade for your presentation is credit-based, but you may lose points if your solution is incorrect or if you take over 10 minutes. The professor may assign someone else to present in your place for bonus, if you miss your presentation or it is incorrect.

We will take time in class to do code reviews or discuss difficulties with completing the worksheets or quizzes from previous classes.

## Examinations

**Midterm exams:** There will be one Midterm exam during the semester.

**Final exam:** One final cumulative exam.

The midterm and final will contain questions similar to the worksheets and assignments and quizzes. Exams are closed book, closed notes, and comprehensive. Exams are in-person, unless there are extraordinary circumstances. The exams should be done individually. No make-up exams except in case of verifiable emergency circumstances. Exams missed will receive zero, unless there is a medical excuse with a physician note.

# AI Policy

Students are expected to complete each assignment without substantive assistance from automated tools (artificial intelligence or machine learning tools such as ChatGPT, DALL-E, Grammarly, GitHub CoPilot, etc.). *Canvas checks submissions for plagiarism from multiple online sources.* Oral examination may be requested.

## ✓ Grading Information

---

Final Grade is based on:

45% Assignments

20% Midterm

30% Final

4% Quizzes

1% Presentation

<i>Grade</i>	<i>Points</i>	<i>Percentage</i>	<i>Interpretation</i>
<i>A plus</i>	<i>960 to 1000</i>	<i>96 to 100%</i>	<i>Exceptional</i>
<i>A</i>	<i>930 to 959</i>	<i>93 to 95%</i>	<i>Excellent</i>
<i>A minus</i>	<i>900 to 929</i>	<i>90 to 92%</i>	<i>Very good</i>
<i>B plus</i>	<i>860 to 899</i>	<i>86 to 89 %</i>	<i>Good</i>
<i>B</i>	<i>830 to 859</i>	<i>83 to 85%</i>	<i>Fair</i>
<i>B minus</i>	<i>800 to 829</i>	<i>80 to 82%</i>	<i>Fair</i>
<i>C plus</i>	<i>760 to 799</i>	<i>76 to 79%</i>	<i>Passed</i>
<i>C</i>	<i>730 to 759</i>	<i>73 to 75%</i>	<i>Passed</i>

<i>C minus</i>	700 to 729	70 to 72%	<i>Barely passed</i>
<i>D plus</i>	660 to 699	66 to 69%	<i>Fail</i>
<i>D</i>	630 to 659	63 to 65%	<i>Fail</i>
<i>D minus</i>	600 to 629	60 to 62%	<i>Fail</i>

## University Policies

---

Per [University Policy S16-9 \(PDF\)](http://www.sjsu.edu/senate/docs/S16-9.pdf) (<http://www.sjsu.edu/senate/docs/S16-9.pdf>), relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on the [Syllabus Information](https://www.sjsu.edu/curriculum/courses/syllabus-info.php) (<https://www.sjsu.edu/curriculum/courses/syllabus-info.php>) web page. Make sure to visit this page to review and be aware of these university policies and resources.

## Course Schedule

---

Lectures	Week	Topic	Chapter
1	1	Introduction: Algorithms & Computers, multiplication of integers	CLRS: Ch 1 & Appendix A
2	1	Review Data Structures (lists, stacks, queues, trees), recursion, basic sorting algorithms	CLRS: Ch 10. SW: Ch 1.3
3	2	Selection Sort, Insertion Sort	CLRS: Ch 2.1. SW: Ch 2.1
4	2	Growth of functions- $O$ , $\Omega$ , $\Theta$ , $o$ , $\omega$	CLRS: Ch 3. SW: Ch 1.4

5	2	Divide and Conquer technique: Merge Sort, Matrix Multiplication	CLRS: Ch 2.2, 2.3. SW: Ch 2.2
6	3	Solving recurrences - Master Theorem & Matrix multiplication revisited	CLRS: Ch 4.3-4.5
7	3	Intro to Heaps, Priority Queues	CLRS: Ch 6.1 SW: Ch 2.4
8	3	Heapsort	CLRS: Ch 6 SW: Ch 2.4
9	4	Graphs, BFS	CLRS: Appendix B.1, B.4-5, Ch 22.1
10	4	DFS	CLRS: Ch 22.2 SW: Ch 4.1-4.2
11	4	Topological sort	CLRS: Ch 22.3-5
12	5	SCC	CLRS: Ch 22.3-5
13	5	Quicksort, Analysis of Quicksort	CLRS: Ch 7 (not 7.3). SW: Ch 2.3
	5	<b>Midterm</b> Monday, July 7, 1:00-3:00pm	

14	6	Order statistics - Selection Algorithm	CLRS: Ch 9 (not 9.2)
15	6	Sorting in linear time, Counting sort, Radix sort, Bucket sort	CLRS: Ch 8



16	6	Binary Search Trees	CLRS: Ch 12 SW: Ch 3.2
17	7	Balanced search trees: 2-3 trees	CLRS: Ch 13 SW: Ch 3.3
18	7	Hashing	CLRS: Ch 11 SW: Ch 3.4
19	7	Union-Find: Data Structures for Disjoint Sets, Union Find, Dynamic sets	CLRS: Ch 12. SW: Ch 1.5
20	8	Minimum Spanning Tree (greedy) – Prim's & Kruskal's Algorithm	CLRS: Ch 23, Ch 21 SW: Ch 4.3
21	8	Single Source Shortest Paths: Dijkstra's Algorithm (greedy), Bellman-Ford introduction (dynamic)	CLRS: Ch 24 SW: Ch 4.4
22	8	Greedy technique	CLRS: Ch 16
23	9	Greedy technique (Activity Selection, knapsack, Huffman codes, scheduling, clustering)	CLRS: Ch 16
24	9	Dynamic Programming technique (Activity Selection, Fibonacci, Bellman-Ford again, All-Pairs Shortest Paths: Floyd-Warshall)	CLRS: Ch 15
25	9	Dynamic Programming (Knapsack, LCS/sequence alignment, optimal search trees, independent set)	CLRS: Ch 15
26	10	NP-completeness, Reductions	CLRS: Ch 34.1-4

27	10	NP-complete problems , Review for exam	CLRS: Ch 34.5 SW: Ch. 6.5
	10	<b>Final exam</b>  Wednesday, August 6, 1:00-3:00pm	