San José State University
College of Science/Computer Science Department
CS146 Sections 4&5, Data Structures and Algorithm, Fall 2022

Course and Contact Information

Instructor(s):	Aikaterini Potika
Office Location:	MacQuarrie Hall 215
Telephone:	408-9245134
Email:	katerina.potika@sjsu.edu
Office Hours:	Tuesdays 1-2 pm & Wednesdays 9:15-10:15 am (online) or by appointment for in person or online Join from PC, Mac, Linux, iOS or Android: https://sjsu.zoom.us/j/91441895686?pwd=Nlp1aExvU2JtaTNKY3VOblk4NEdjQT09 Password: 793531
Class Days/Time:	Mondays-Wednesdays 12:00-1:15 pm (Section 04) and 1:30-2:45pm (Section 5)
Classroom:	MacQuarrie Hall 222
Prerequisites:	MATH 30, MATH 42, and (CS 46B in Java or (CS 49J and CS 46B)) (with a grade of "C-" or better in each); or instructor consent.

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 Format

Technology Intensive, Hybrid, and Online Courses

The course will be an online hybrid course. For the CS Department it is online synchronous zoom meetings plus recording lectures, thus allowing students to listen to recorded lecturers asynchronously.

Faculty Web Page and MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on my faculty on Canvas Learning Management System course login website at http://sjsu.instructure.com. You are responsible for regularly checking with the messaging system through MySJSU on Spartan App Portal http://one.sjsu.edu

(or other communication system as indicated by the instructor) to learn of any updates. For help with using Canvas see Canvas Student Resources page (http://www.sjsu.edu/ecampus/teaching-tools/canvas/student_resources)

Course Learning Outcomes (CLO)

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

- CLO1. 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
- CLO2. Prove basic properties of trees and graphs
- CLO3. Perform breadth-first search and depth-first search on directed as well as undirected graphs
- CLO4. Use advanced sorting techniques (heapsort, mergesort, quicksort)
- CLO5. Determine the running time of an algorithm in terms of asymptotic notation
- CLO6. Solve recurrence relations representing the running time of an algorithm designed using a divide-and-conquer strategy
- CLO7. 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
- CLO8. Understand algorithms designed using greedy, divide-and-conquer, and dynamic programming techniques

Required Texts/Readings

Textbook

Cormen, Leiserson, Rivest and Stein, Introduction to Algorithms, 3rd Edition
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.

Other Readings

- Horstmann and Cornell, Core Java, Vol. I, Ninth edition, Prentice Hall, 2013.
- Kleinberg and Tardos, Algorithm Design, First edition, Addison Wesley, 2005.
- Dasgupta, Papadimitriou and Vazirani, Algorithms, McGraw-Hill, 2006.

Other technology requirements / equipment / material

Java Compiler (version 8 or later).

Course Requirements and Assignments

“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.

Homework assignments: individual, regularly assigned, not graded, include written problem assignments, and perhaps some online exercises. Solutions are not posted. The homework is a tool for you to learn the material and prepare for the exams.

Reading and Video assignments: Reading assignments and posted videos are regular and for the next class (see schedule).

Quizzes: Weekly quizzes are online or offline. Cover topics from the reading and video assignment and/or the homework. Various in class activities.

Participation & Discussions: Contribution during zoom meetings, polls and activities, and in the discussion forum of Canvas.

Programming assignments: Programming assignments are assigned. Programming assignments are done individually, unless otherwise specified. They can be discussed but should be implemented individually. More information is given at the time of the first programming assignment. Never use any code you find on the web, unless I provide it. Some assignments have an oral examination.

Midterm exam: One Midterm exam during the semester.

Final Examination or Evaluation: One final, written, and cumulative exam, split in two parts. The exams contain multiple-choice questions, short answer questions and questions that require pseudocode and/or computations.

Grading Information

No extra point options. Final exam is comprehensive.

Grading Information

Determination of Grades

No make-ups exams except in case of verifiable emergency circumstances. Penalty for late submission, 5% for every 3 days up to 9 days, after that no submission is accepted (without counting weekends). Never email your assignments, always upload to Canvas. Rubrics and examples will be given.

Final Grade:

15% Programming assignments

20% Quizzes

5% Participation (Polls, Activities)

5% Discussions

20% Midterm

35% Final

Grade	*Percentage*
A plus	96 to 100%
A	93 to 95%
A minus	90 to 92%
B plus	86 to 89 %
B	82 to 85%
B minus	78 to 82%
C plus	74 to 77%
C	70 to 73%
C minus	65 to 69%
D plus	62 to 64%
D	58 to 61%
D minus	55 to 57%
F	<54%

Embedded Tutoring

Embedded tutoring is a form of supplemental instruction offered by Peer Connections in which the tutor attends class meetings and, under the instructor's guidance, helps students understand concepts and contribute to class discussions. The tutor also holds weekly "office hours" that students are encouraged to attend. For more information, please visit http://peerconnections.sjsu.edu/

Tutor: Saharsh Vedi <saharsh.vedi@sjsu.edu>

Extra points: If you attend one session of tutoring until Midterm you will receive extra 2% in the end.

Classroom Protocol

During zoom meetings: camera on, mute yourself (unless you have a question or want to contribute), and dress appropriately. Private interactions with other students are prohibited unless you are in a breakroom. Please avoid disturbing the class: turn-off cell phones (or put them on vibrate mode), no text messaging in the class or the exams, no taking pictures and video, avoid coming late, no talking or whispering with other students during

the instructor's presentation. You may not publicly share or upload material of this course such as exam questions, lecture notes, or solutions without my consent.

University Policies

Per University Policy S16-9 *(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 Syllabus Information web page (https://www.sjsu.edu/curriculum/courses/syllabus-info.php). Make sure to visit this page to review and be aware of these university policies and resources. Add/drop deadline 9/15.

The instructor reserves the right to drop students that do not show up during the first two lectures.

COVID-19 and Monkeypox Safety Training

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).

CS146: Data Structures and Algorithms, Fall 2022, Course Schedule

The schedule is subject to change with fair notice and announced on Canvas.

Course Schedule

Lesson	Date	Topic	Reading/Projects
1	8/22	Introduction: Algorithms & Computers	Ch 1 & Appendix A
2	8/24	Review recursion, Data Structures (lists, stacks, queues, trees)	Ch 10 Project 1
3	8/29	Insertion Sort	Ch 2.1
4	8/31	Growth of functions- O, Ω, Θ, o, ω	Ch 3
5	7/9	A Survey of Common Running Times	Ch 3

6	9/12	Divide and Conquer technique: Merge Sort	Ch 2.2, 2.3
7	9/14	Divide and Conquer technique: other examples	Ch 2.2, 2.3 Project 2
8	9/19	Solving recurrences - Master Theorem	Ch 4.3-4.5
9	9/21	Master Theorem	Ch 4.3-4.5
10	9/26	Intro to Heaps	Ch 6.1
11	9/28	Heapsort, Priority Queues	Ch 6
12	10/3	Graphs	Appendix B.1, B.4-5
13	10/5	Breadth First Search	Ch 22.1
14	10/10	Depth First Search, Topological sort	Ch 22.2, Ch 22.3-5 Project 3
	10/12	**Midterm**	
15	10/17	Analysis of Quicksort	Ch 7 (not 7.3)
16	10/19	Order statistics - Selection Algorithm	Ch 9 (not 9.2)
17	10/24	Sorting in linear time, Counting sort, Radix Sort	Ch 8
18	10/26	Sorting in linear time, Counting sort, Radix Sort	Ch 8
19	10/31	Searching: Hashing	Ch 11
20	11/2	Searching: Binary Search Trees, Red Black trees	Ch 12, Ch 13
21	11/7	Searching: Red Black trees	Ch 13 Project 4

22	11/9	Intro to Greedy technique	Ch 16
23	11/14	Intro to Dynamic Programming	Ch 15
24	11/16	Greedy technique examples	Ch 16
25	11/21	Single Source Shortest Paths: Dijkstra's Algorithm	Ch 24
26	11/28	Minimum Spanning Tree - Prim's and Kruskal's Algorithm, Data Structures for Disjoint Sets	Ch 23, Ch 21
27	11/30	Dynamic Programming technique examples	Ch 15, 25
28	12/5	NP-complete problems	Ch. 34.1-4
		Final exam Wednesday, December 14, 9:45 AM-12:00 PM (Section 4) Tuesday, December 13, 12:15-2:30 PM (Section 5)	