San José State University
Computer Science Department
CS146, Data Structures and Algorithms, Section 1, Spring 2020

Course and Contact Information

Instructor(s):	Aikaterini Potika
Office Location:	MacQuarrie Hall 215
Telephone:	408-9245134
Email:	katerina.potika@sjsu.edu
Office Hours:	T 1:30-3pm and Th 9:45-10:15am am or by appointment
Class Days/Time:	TTh 10:30-11:45 am
Classroom:	MacQuarrie Hall 222
Prerequisites:	MATH 030, MATH 042, CS 049J (or equivalent knowledge of Java), and/or CS 046B (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

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 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 7 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 be posted. The homework is a tool for you to learn the material and prepare for the exams.

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

Quizzes: Unannounced quizzes (at least 4) may be given during class, each taking about 5 minutes total. These generally are problems from the reading assignment and/or the homework.

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. *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. Oral examination might be requested.

Midterm exams: Two written Midterm exams 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 (only the final exam offers extra points option).
All exams are closed book, and final exam is comprehensive. No make-ups exams except in case of verifiable emergency circumstances.

Determination of Grades

Final Grade:
15% Programming assignments
5% Quizzes
40% Midterms (20% each)
40% Final

A+	A	A-	>90
B+	B	B-	>78
C+	C	C-	>65
D+	D	D-	>45
F			<40

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: Maan Singh <maan.singh@sjsu.edu> (Section 1)
Extra points: If you attend one 30' of tutoring until Midterm 1 you will receive extra 2% in the end.

Classroom Protocol

Attendance is highly recommended. 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 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 (http://www.sjsu.edu/gup/syllabusinfo), which is hosted by the Office of Undergraduate Education. Make sure to visit this page to review and be aware of these university policies and resources.

The instructor might drop students that do not show up during the first two lectures.

CS146: Data Structures and Algorithms, Spring 2020

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

Course Schedule

Lesson	Date	Topic	Reading/Projects
1	1/23	Introduction: Algorithms & Computers	Ch 1 & Appendix A
2	1/28	Review Data Structures (lists, stacks, queues, trees), recursion, basic algorithms	Ch 10, Project 1
3	1/30	Insertion Sort	Ch 2.1
4	2/4(drop)	Growth of functions- O, Ω, Θ, o, ω	Ch 3 (2 lectures)
5	2/6	Growth of functions- O, Ω, Θ, o, ω	
6	2/11 (add)	Divide and Conquer technique: Merge Sort other examples	Ch 2.2, 2.3, Project 2
7	2/13	Solving recurrences - Master Theorem	Ch 4.3-4.5
8	2/18	Master Theorem	Ch 4.3-4.5
9	2/20	Intro to Heaps	Ch 6.1
10	2/25	Heapsort, Priority Queues	Ch 6
	2/27	**Midterm 1**	
11	3/3	Graphs	Appendix B.1, B.4-5, Project 3
12	3/5	BFS	Ch 22.1
13	3/10	DFS, Topological sort	Ch 22.2, Ch 22.3-5
14	3/12	Quicksort, Analysis of Quicksort	Ch 7 (not 7.3)
15	3/17	Order statistics - Selection Algorithm	Ch 9 (not 9.2)

16	3/19	Sorting in linear time, Counting sort, Radix Sort	Ch 8
17	3/24	Union Find, Dynamic sets, Binary Search Trees	Ch 12
18	3/26	Red Black trees	Ch 13, Project 4
	3/31-4/2	*Spring break*	
19	4/7	Intro to Greedy technique and Dynamic Programming	Ch 15, 16 (2 lectures)
20	4/9	Intro to Greedy technique and Dynamic Programming	
21	4/14	Hashing	Ch 11
	4/16	Midterm 2	
22	4/21	Minimum Spanning Tree – Prim's and Kruskal's Algorithm, Data Structures for Disjoint Sets	Ch 23, Ch 21
23	4/23	Single Source Shortest Paths: Dijkstra's Algorithm	Ch 24
24	4/28	Dynamic Programming technique	Ch 15
25	4/30	Dynamic Programming technique examples	Ch 15, 25
26	5/5	NP-complete problems	Ch. 34.1-4
27	5/7	NP-complete problems	Ch. 34.1-4
		Final exam Monday, May 18 | 09:45-12:00	