San José State University
Computer Science Department
CS146, Section 1&2, Data Structures and Algorithms, Fall 2016

Course and Contact Information

Instructor:	Aikaterini Potika
Office Location:	MacQuarrie Hall 215
Telephone:	408-9245134
Email:	katerina.potika@sjsu.edu
Office Hours:	T 1:30-3:30 or by appointment
Class Days/Time:	TTh 10:30-11:45am (Section 1) 12:00-13:15pm (Section 2)
Classroom:	Duncan Hall 450
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 Format

Faculty Web Page and MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on Canvas Leaning Management System course login website at http://sjsu.instructure.com. You are responsible for regularly checking with the messaging system through MySJSU at http://my.sjsu.edu (or other communication system as indicated by the instructor) to learn of any updates.

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 Learning Outcomes (CLO)

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

1. 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
2. CLO2. Prove basic properties of trees and graphs
3. CLO3. Perform breadth-first search and depth-first search on directed as well as undirected graphs
4. CLO4. Use advanced sorting techniques (heapsort, mergesort, quicksort)

5. CLO5. Determine the running time of an algorithm in terms of asymptotic notation
6. CLO6. Solve recurrence relations representing the running time of an algorithm designed using a divide-and-conquer strategy
7. 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
8. 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.
- Handouts (through Canvas)

Other technology requirements / equipment / material

Java Compiler (version 7 or later).

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.

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

Reading assignments: Reading assignments will regularly be for the next class (see schedule).

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

Programming assignments: Programming assignments will be assigned. Programming assignments are done individually, unless otherwise specified. They can be discussed, but should be implemented individually. More information will be given at the time of the first programming assignment. Never use any code you find on the web, unless I give it. Penalty for late submission 5% for every 3 days up to 9 days, after that no submission will be accepted. Never email your assignments, always upload to Canvas. Oral examination might required.

Additionally, you may be disqualified from getting an A/B/C/D unless you get at least 4/3/2/1 program(s) to work, respectively.

Midterm exams: There will be two written Midterm exams during the semester.

Final Examination or Evaluation

Final exam: One written final exam.
The exams will contain multiple choice questions, short answer questions and questions that require pseudocode and/or computations.

Grading Information

Students must obtain >50% in each component of the course (programming & quizzes & written exams) in order to be eligible for a grade of C- or better. No extra point options.

Determination of Grades

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

Exams are closed book, final exam is comprehensive. No make-ups exams except in case of verifiable emergency circumstances

A+	A	A-	>90
B+	B	B-	>75
C+	C	C-	>60
D+	D	D-	>45
F			<40

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. You may not publicly share or upload material for this course such as exam questions, lecture notes, or solutions without my consent.

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/"

CS146: Data Structures and Algorithms, Fall 2016

The schedule is subject to change with fair notice.

Course Schedule

Lectures	Date	Topic	Chapter
1	8/25	Introduction: Algorithms & Computers	Ch 1 & Appendix A
2	8/30	Review Data Structures (lists, stacks, queues, trees) & Insertion Sort	Ch 10, Ch 2.1
3	9/1	Growth of functions- O, Ω, Θ, o, ω	Ch 3
4	9/6	Divide and Conquer technique: Merge Sort & Matrix multiplication	Ch 2.2, 2.3
5	9/8	Solving recurrences - Master Theorem	Ch 4.3-4.5
6	9/13	Master Theorem - Intro to Heaps	Ch 6.1
7	9/15	Heapsort, Priority Queues	Ch 6
8	9/20	Quicksort	Ch 7 (not 7.3)
9	9/22	Analysis of Quicksort	Ch 7 (not 7.3)
10	9/27	Review	
	9/29	**Midterm 1**	
11	10/4	Sorting in linear time, Counting sort, Radix Sort,	Ch 8
12	10/6	Order statistics - Selection Algorithm	Ch 9 (not 9.2)
13	10/11	Hashing	Ch 11
14	10/13	Union Find, Dynamic sets, Binary Search Trees	Ch 12
15	10/18	Red Black trees	Ch 13
16	10/20	B-trees	Ch 18
17	10/25	Greedy technique	Ch 16
18	10/27	Dynamic Programming technique	Ch 15
19	11/1	Graphs	Appendix B.1, B.4-5

20	11/3	BFS	Ch 22.1,
20	11/8	DFS, review	Ch 22.2
21	11/10	**Midterm 2**	
22	11/15	Topological sort	Ch 22.3-5
23	11/17	Minimum Spanning Tree – Prim's and Kruskal's Algorithm, Data Structures for Disjoint Sets, Review	Ch 23, Ch 21
24	11/22	Single Source Shortest Paths: Dijkstra's Algorithm	Ch 24
25	11/29	All-Pairs Shortest Paths: Floyd-Warshall	Ch 25.1-2
26	12/1	NP-completeness, Reductions	Ch. 34.1-4
27	12/6	NP-complete problems	Ch. 34.5
28	12/8	Review	
		Final exam Section 1: Th 12/15 9:45-12pm Section 2: M 12/19 9:45-12pm	