San Jose State University
Computer Science Department

Course Syllabus
CS146, Data Structures and Algorithms
Fall 2019

Course Number:	CS146
Course Title:	Data Structures and Algorithms
Credits:	3
Class hours:	Tuesdays/Thursdays 1:30pm to 2:45pm
Class location:	Science building, room 311
Instructor:	Maryam Hasan (Maryam.hasan@sjsu.edu)
Office hours:	Tuesdays 2:45pm to 3:45pm
Prerequisite:	MATH 42, MATH 30, CS 46B, CS 49J (or equivalent knowledge of Java) (with a grade of "C-" or better in each).
Textbook:	Introduction to Algorithms, 3rd Edition Cormen, Leiserson, Rivest, and Stein ISBN-10: 0262033844ISBN-13: 978-0262033848MIT Press, 2009 https://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844
Course Website:	The course material will be available through canvas.

Faculty Web Page and MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found at Canvas of SJSU One. You are responsible for regularly checking with the email system and Canvas through One.SJSU at http://one.sjsu.edu 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, merge sort, and quicksort). Design and analysis of data structures and algorithms. Divide-and-conquer, greedy, and dynamic programming algorithm design techniques.

Course Objectives

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

Course Learning Outcomes (CLO)

Upon successful completion of this course, students should 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 Policies:

Attendance: Attendance is mandatory.

Homework assignments: Most homework will be due before class time on the due date. Late homework will not be accepted.

Exams: There are two exams in this course, midterm and final. If you miss the exam you cannot pass this class. Rescheduling an exam will only be allowed in highly selective and pre-approved cases. You need to contact the instructor at least 2 weeks in advance for approval. If the scheduled exam dates are in conflict with your religious observances, you must notify the instructor, in writing, at least two weeks in advance of the exam.

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 (Required)

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/" Make sure to review these policies and resources.

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. More details about student workload can be found in University Policy S12-3at http://www.sjsu.edu/senate/docs/S12-3.pdf.

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

Grading information:

Homework assignments: 40%

In-class pop-up quizzes/tests: 10%

Midterm exam: 25%

Final exam: 25%

The final letter grade for the course will be calculated using the following distribution:

Percentage	Grade
95 and above	A+
92-94	A
90 - 91	A-
87 - 89	B+
83 - 86	B
80 - 82	B-
77 - 79	C+
73 - 76	C
70 - 72	C-
67 - 69	D+
63-66	D
60-62	D-
59 and below	F

Tentative Schedule**:**

Session	**Topic**	**Homework Assignments**
1. August 22	Syllabus, Introduction and Basic Concepts	
2. August 27	Growth of Functions- O, Ω, Θ	
3. August 29 4. September 3 5. September 5	Elementary Data Structures *stacks, queues, hash tables, trees, binary search trees*	
6. September 10	Elementary Data Structures *B+tree, heaps, graphs*	HW1
7. September 12		
7. September 17		
8. September 19	Recursive algorithms *Substitution method* *Master theorem*	
9. September 24		
10. September 26	Sorting algorithms insertion sort, selection sort,	
11. October 1		
12. October 3	Sorting algorithms heap sort, merge sort, quicksort	HW2
13. October 8		
14. October 10		
15. October 15	Design approaches divide & conquer	
16. October 17		
17. October 22	Midterm exam	
18. October 24	Design approaches dynamic programming	

19. October 29	Design approaches dynamic programming	
20. October 31	Design approaches greedy algorithms	HW3
21. November 5 22. November 7	Graph algorithms depth-first search, breadth-first search,	
23. November 12	Graph algorithms shortest path: Dijkstra's algorithm	
24. November 21	Graph algorithms shortest path: Bellman-Ford	
25. November 26	Graph algorithms minimum spanning tree	HW4
26. November 28	No class- Thanksgiving	
27. December 3	NP-complete problems	
28. December 5	Final Review	
29. December 12	Final Exam	