SP23: CS-185A Sec 01 - Adv Prac Comp Topics

✎ Edit

San José State University

College of Science/Computer Science Department

CS 185A, Applied Algorithms II

Course and Contact Information

Instructor(s): Ben Reed

Office Location: MH 213

Telephone: (408) 924-5174

Email: ben.reed@sjsu.edu

Office Hours:

4:30-6:00PM Monday

10-11:30AM Tuesday over zoom https://sjsu.zoom.us/j/4077267356

11:30-1PM Wednesday

3:30-5PM Thursday over zoom https://sjsu.zoom.us/j/4077267356
Links to an external site.

Class Days/Time: Wednesdays 4:30-5:20PM

Classroom: Clark 111

Prerequisites: CS 146

Course Description

1 unit. Applying data structures and algorithms from CS 146 to implement programs and solve problems. Students will be expected to write a program every week.

Course Format

This is an in-person course.

Course Learning Outcomes (CLO)

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

1. demonstrate proper application of algorithms and techniques such topological sort, flows, and dynamic programming.
2. evaluate when basic data structures such as arrays, linked lists, hash tables, trees, and heaps should be used.
3. develop solutions to common programming problems used in industry to test advanced programming ability.
4. explain solutions to problems and implement them in code.
5. describe the space/time complexity of program examples.
6. implement an abstract description of a solution in code.

Required Texts/Readings

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

Course Requirements and Assignments

This course will ensure that you are able to apply the concepts you have learned is CS 146. We do this by writing lots of code! Don’t worry, it will be FUN! We will be using concepts you have already learned from 146. Every week we will pick a new problem to solve. You will work on the solution BY YOURSELF. The next class we will go over the solution to the problem and start a new one. The key is that you practice your ability to write code to apply concepts that you have learned and thereby develop confidence in your programming ability.

Programming assignments (70%)

We will be doing individual programming assignments. We will review the solution the class after the assignment is due. The assignments are due before class, but you may submit them two days late with 20 points off. You will submit your assignment to sjsu.kattis.com. Since unexpected events may arise, two of the weekly assignments can be dropped without penalty.

An example of a problem for week two is **https://open.kattis.com/problems/heimavinna** **(https://open.kattis.com/problems/heimavinna)** .

Individual programming assignments are not group projects. If students get help on assignments, even to resolve a stupid problem, it must be documented in the code with the name of the person rendering the help and a brief description of the help provided. You will lose points if you get help. Extensive help on a project will disqualify the submission.

Failure to document help, or any other forms of cheating will result in a failing grade on the assignment at a minimum and may result in failure of the course. All incidents will be reported

to the Office of Student Conduct & Ethical Development. Even in open source, you cannot copy code from one open source project to another without attribution. Sharing solutions with other students, even if it is indirectly through public source repositories, falls under "aiding and abetting".

The University Policy S16-9, Course Syllabi (http://www.sjsu.edu/senate/docs/S16-9.pdf) requires the

following language to be included in the syllabus:

“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

By the middle of the semester, the hope is that you will be comfortable enough coding that you will be able to solve each weeks problem in less than an hour.

Final Examination and in class programming problems (30% total)

There will be two programming problems that will be done in class. They will be timed and will be administered in the style of an exam. You will be able to use only language references. The problems are chosen to be similar to previous weekly assignments.

Grading Information

This class uses minimum grading: you cannot get below a 50% on any submission or exam. For example, if you do not submit a solution explanation or your submission falls far short and only scores 35%, you will be assigned a 50% in the grade book. The minimum grading does not apply to cases of academic integrity.

programming assignments	70%
in class programming problem	30%

Grade	*Percentage*
Credit	*75 to 100%*

Classroom Protocol

This is your class. Please ask questions. Please come prepared. Do not engage in activity that may distract other students.

I do not take attendance except for the first two classes. Students not attending either of the first two classes will be dropped to make room for students on the waiting list. Attempting to get marked as present (by having someone else attend in your place or using technological deceptions) will be considered academic dishonesty and at a minimum will result in you getting dropped from the course.

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)** (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.

Course Schedule

Week/Lesson/Module	**Topics, Readings, Assignments, Deadlines**
1	Recursive exploring
2	Dynamic Programming with caching
3	Dynamic Programming with precalculation
4	DFS
5	BFS

Week/Lesson/Module	Topics, Readings, Assignments, Deadlines
6	In-class problem 1
7	Topological Sort
8	Connected Components
9	Flows
10	Spring break
11	In-class problem 2
12	Modulo
13	Math
14	Strings
15	Strings
16	Wrapup
Final	In-class problem (Wednesday 5/17/2023 @ 2:45PM)