

San José State University
College of Science/Computer Science Department
CS 85A, Applied Algorithms

Course and Contact Information

Instructor(s): Ben Reed
Office Location: MH 213
Telephone: (408) 924-517
Email: ben.reed@sjsu.edu
Office Hours: TBD
Class Days/Time: TBD
Classroom: TBD
Prerequisites: CS 46B

Course Description

1 unit. Applying basic data structures to implement programs to solve problems. Emphasis on using the data structures learned in CS 46B. 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 usage of APIs for data input and output.
2. evaluate the when basic data structures such as arrays, linked lists, and hash tables should be used.
3. develop solutions to common programming problems used in industry to test programming ability.
4. explain solutions to problems and implement them in code.
5. analyze potential performance problems in code and device solutions improve performance.
6. implement an abstract description of a solution in code.

Required Texts/Readings

Course Requirements and Assignments

This course will ensure that you are able to apply the concepts you have learned in CS 46B. 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 46B. 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 (60%)

We will be doing individual programming assignments. THERE ARE NO LATE ASSIGNMENTS. We will review the solution the class after the assignment is due. You will submit your assignment to canvas after

making it pass on open.kattis.com or leetcode.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> . You will be able to implement your solution and get it tested before submitting it to Canvas.

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. Extensive help on a project will disqualify the submission.

The one exception to this rule is one non-weekly assignment to solve a problem of your choice using the method presented in the video of your peer (see “Solution explanation”).

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.

Solution explanation (10%)

You can select a problem to explain and solve in a video. Students should be able to implement your solution after watching the video. You cannot make extensive use of code in the video; you need to focus on explaining the key ideas of the solution.

IDE Tip video (10%)

IDEs make us much more productive as programmers. In this class, you can use the IDE of your choice. To help others learn to be more effective with IDEs, you will create a short video (3-5 mins) that show an awesome feature of your favorite IDE.

Final Examination and in class programming problem (20% 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	60%
solution explanation	10%
IDE tip video	10%
in class programming problem	20%

Grade	Percentage
Credit	70 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](#), 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 (Delete if not applicable)	Topics, Readings, Assignments, Deadlines (If appropriate, add extra column(s) to meet your needs.)
1	Input/Output
2	Input/Output
3	Strings
4	Math/bit operations
5	Arrays

Week/ Lesson/ Module (Delete if not applicable)	Topics, Readings, Assignments, Deadlines (If appropriate, add extra column(s) to meet your needs.)
6	Arrays
7	Arrays/searching
8	Arrays/sorting
9	Linked Lists
10	Hashtables
11	Hashtables
12	Simple big O
13	In-class problem
14	Bit vectors
15	Lambdas
16	Finding performance problems
Final	In-class problem