SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Applied Algorithms

CS 48

Spring 2026 Section 01 In Person 1 Unit(s) 01/22/2026 to 05/11/2026 Modified 01/22/2026

Contact Information

Instructor: Chase Potter

Email: chase.potter@sjsu.edu

Course-related correspondence should generally go to Chase Potter, the student instructor. He will be your main contact for the course.

David Scot Taylor

Email: david.taylor@sjsu.edu

Office Hours

Monday, 12:00 PM to 2:00 PM, MacQuarrie Hall 212

How little should you expect to see me? I am teaching another class at the same time that this course meets. Chase will really be your contact for the course, and I will be in communication with him. However, you can come to my office hours if you want to.

I will not have office hours on March 2.

Course Information

Lecture

Monday, 9:00 AM to 9:50 AM, MH 424

Course Description and Requisites

Creating and implementing algorithms to solve problems. Techniques covered include using built-in collection classes, bitwise operators, modulo operator, and input/output classes. Emphasis on using data structures learned in CS 46B. Students write a Java program every week.

Prerequisite(s): CS 46B and one Java course (with grades of C- or better), or instructor consent.

Letter Graded

Classroom Protocols

Note, this is an in-person class.

Program Information

Diversity Statement - At SJSU, it is important to create a safe learning environment where we can explore, learn, and grow together. We strive to build a diverse, equitable, inclusive culture that values, encourages, and supports students from all backgrounds and experiences.

Course Goals

This course intends to help you to apply the concepts you have learned is CS 46B. We do this by writing lots of code---every week we will pick a new problem to solve, using Java. 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.

Course Learning Outcomes (CLOs)

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.

Course Materials

Students should have a laptop available, with Java and an IDE installed. While students are free to choose their IDE, their IDE should at least have a debugger.

Course Requirements and Assignments

Programming assignments (55%)

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 a system called WebCAT. Since unexpected events may arise, two of the weekly assignments can be dropped without penalty. You will be able to submit your code to WebCAT multiple times.

Individual programming assignments are not group projects. If students get help on assignments, even to resolve a minor 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.

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"

IDE Tip Presentation (10% or 5% + 5%)

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 make a 3-5 minute presentation to show an awesome feature of your favorite IDE. Students can choose their topic. Past topics include triggered and conditional breakpoints, advanced source control, automated refactoring tools, step filters, logpoints/tracepoints, debug consoles, and data breakpoints.

This assignment ***might*** be broken into two parts: creating your own video, and questions related to watching videos of other students. This will be determined later in the semester.

Program explanation (5%)

Shortly after the midterm, each student will be expected to explain a solution to a programming problem of their choice to the class. Your explanations should explain how your solution works without simply showing your solution's code. A good explanation will leave the audience with a clear picture of how to implement your solution in code.

Final Examination and in-class programming problem (30%)

There will be three 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 limited to only use language references (the Oracle Java docs) for these. The problems are chosen to be similar to weekly assignments.

The first problem will be the midterm. Students will have the entirety of class to write a fully-functioning solution and submit it to WebCAT.

The second and third problems are the final. Students will have the full final period to write their fully-functioning solutions and submit them to WebCAT. The programs are graded individually.

✓ Grading Information

This is a graded class, but we will use 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	55%
IDE tip video	10%
solution explanation	5%
in class programming problems	30%

Your course letter grade will be determined by your final weighted average according to the below table (standard letter grades):

A+	>= 97%
A	>= 93% and < 97%
A-	>= 90% and < 93%
B+	>= 87% and < 90%
B	>= 83% and < 87%
B-	>= 80% and < 83%
C+	>= 77% and < 80%
C	>= 73% and < 77%
C-	>= 70% and < 73%

D+	>= 67% and < 70%
D	>= 63% and < 67%
D-	>= 60% and < 63%
F	>= 0% and < 60%

There is no rounding, but boundary cases count as the higher of the two grades. The instructor reserves the right to make cutoffs more lenient, but not more strict.

University Policies

Per University Policy S16-9 (PDF) (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 the Syllabus Information (https://www.sjsu.edu/curriculum/courses/syllabus-info.php) web page. Make sure to visit this page to review and be aware of these university policies and resources.

Course Schedule

When	Topic	Notes
1/26/26	IO	Scanner, BufferedReader, FileIO
2/02/26	Strings and Tests	StringBuilder, String methods, String memory allocation JUnit
2/09/26	Binary Search Trees	Insertion, Deletion, Find Postorder, Preorder, Inorder traversals
2/16/26	Math and Bit Operations	xor, or, and, not, shift left, shift right

2/23/26	Arrays	Arrays vs ArrayLists Array access Boxing/Unboxing Array methods Applications (prefix sum, etc)
3/02/26	Searching	Binary Search implementation Unbounded Binary Search Binary search for closest element
3/09/26	Sorting	Builtin sorts Comparable/Comparator Applications
3/16/26	Midterm In Class Problem (attendance required)	1 problem, all of class
3/23/26	Linked Lists	Memory layout Insertion/deletion time Random access time Algorithms on linked lists
3/30/26	No Class!	Spring Break this week.

4/06/26 + 4/13/26	Hashtables and Ordered Maps	Hash Functions Chaining Hashtable or Array? Types of Maps
4/20/26	Big O notation	Definition Complexity Classes and Common Operations Analyzing Programs
4/27/26	Useful Java Data Types	BigInteger, Bitset, BigDouble applications
5/04/26	Lambdas, Functions, and streams	Function, Lambda, Consumer, BiConsumer, BiFunction, Stream, …
5/11/26	Review	In-class Q/A to prepare for final
5/15/26	Final Exam	Two problems. Friday, May 15 8:30 AM to 10:30 PM regular class location