SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Data Structures and Algorithms

CS 146

Fall 2026 In Person 3 Unit(s) 08/19/2026 to 12/07/2026 Modified 08/21/2026

Welcome to CS 146. This course covers the design and analysis of algorithms and data structures, using Java as the implementation language. The concepts here are the foundation for building CS systems and for your career. They carry directly into not just future courses (like databases, operating systems, machine learning, and AI), but also into technical interviews and the rest of your work as an engineer.

Contact Information

- **Instructor:** Dr. Preethi Natarajan
- **Email:** preethi.natarajan@sjsu.edu ()
- **Class Info**: MW 3:00PM - 4:15PM @ MacQuarrie Hall 424
- **Preferred method of contact:** Canvas Inbox

Course Description and Requisites

Implementations of advanced tree structures, priority queues, heaps, directed and undirected graphs. Advanced searching and sorting techniques (radix sort, heapsort, mergesort, and quicksort). Design and analysis of data structures and algorithms. Divide-and-conquer, greedy, and dynamic programming algorithm design techniques.

Prerequisite(s): MATH 30, MATH 42, CS 46B, and [(CS 48 or CS 49J) if CS 46B was not in Java], each with a grade of C- or better; Computer Science, Applied and Computational Math, Forensic Science: Digital Evidence, Software Engineering, Data Science, Computer Science and Linguistics majors only; or instructor consent.
Grading: Letter Graded

Classroom Protocols

Communication with the Instructor
Please use the instructor's university email for course-related questions or concerns. Email is the best way to reach the instructor.

Course Materials and Intellectual Property
Course materials—including lecture slides, notes, presentations, exam questions, programming assignments, exercises, and homework solutions—are intended for students enrolled in this course. Please do not record, reproduce, upload, or share these materials without

the instructor's permission. These guidelines help protect student privacy and maintain a fair and academically honest learning environment.

Unauthorized recording or distribution may be referred to the appropriate university office and may also violate university or state law. Students who require accommodations or assistive technology should contact the instructor so appropriate arrangements can be made.

Attendance and Arrival
Attendance is not mandatory and will not, by itself, affect your grade. However, regular attendance and participation are strongly encouraged. Please arrive on time and be ready to begin when class starts.

Respectful Classroom Environment
Please treat everyone in the classroom with respect. Questions, discussions, and different perspectives are welcome, but please keep conversations constructive and avoid behavior that disrupts others' learning.

Electronic Devices
Laptops, tablets, and smartphones should primarily be used for course-related activities. Please silence cell phones before class and minimize non-course-related device use.

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 Learning Outcomes (CLOs)

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

1. Implement core data structures, including lists, stacks, queues, search trees, heaps, hash tables, union-find, red-black trees, B-trees, and graphs, and apply them in programs they design.
2. Determine the running time of an algorithm in terms of asymptotic notation and analyze the complexity of different algorithms.
3. Prove basic properties of trees and graphs.
4. Perform breadth-first search and depth-first search on directed and undirected graphs.
5. Use advanced sorting techniques, including radix sort, heapsort, mergesort, and quicksort, and analyze their complexity.
6. Solve recurrence relations representing the running time of divide-and-conquer algorithms.
7. Design and analyze algorithms using greedy, divide-and-conquer, and dynamic programming techniques.
8. Comprehend the basic concept of NP-completeness and recognize that not all problems encountered in practice can be solved efficiently.
9. Select and apply efficient data structures and algorithms to solve practical problems.

Course Materials

Textbook

Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms*, 3rd Edition
MIT Press, 2009

- ISBN-10: 0262033844
- ISBN-13: 978-0262033848

Technology Requirements

Students will need access to a **laptop with reliable internet connectivity** throughout the course. The laptop should run macOS, Windows, or a version of UNIX/Linux and allow you to install the required software.

The course will use:

- **Canvas** for course materials, announcements, and assignments
- **Java** for programming assignments
- An **Integrated Development Environment (IDE)** for writing and running code

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 S12-3 (http://www.sjsu.edu/senate/docs/S12-3.pdf)

Course Prerequisites

You must provide documentation showing that you have satisfied the course prerequisites. Please submit proof of the required prerequisites by end of first week. If you do not provide the required documentation by the deadline, you may be dropped from the course if other students are waiting for a seat. In any case, students who have not demonstrated that they have satisfied the prerequisites by the drop date will be dropped from the course.

Please note that I will not issue an add code without first verifying that the prerequisites have been satisfied.

The required prerequisites are:

- Math 30
- Math 42
- CS 46B in Java, **or** CS 49J and CS 46B

You must have earned a grade of **C or better** in each prerequisite. Please provide documentation showing your grades for CS 46B, Math 30, and Math 42, or approved equivalents documented on a San José State departmental course equivalence form.

If you completed CS 46B at another institution, you must also provide either documentation showing that you completed CS 49J or documentation confirming that the CS 46B course you completed was taught in Java.

Grading Information

Assessment Policy

Attainment of the learning objectives will be assessed via several components distributed across the semester. The following breakdown outlines the relative weight of each component toward the final course grade.

- **Quizzes:** 2 quizzes, weighted at 15% of the final grade
- **Homeworks / Assignments:** ~5 assignments, weighted at 35% of final grade
- **Midterms:** 2 midterms, weighted at 30% of final grade
- **Final Exam:** 1 final, weighted at 20% of final grade

Final grades may be curved (up) to raise grades if needed.

Your course grade will be determined by your final weighted average:
A plus = 97% or higher
A = 93% to 97%
A minus = 90% to 93%
B plus = 87% to 90%
B = 83% to 87%
B minus = 80% to 83%
C plus = 77% to 80%
C = 73% to 77%
C minus = 70% to 73%
D plus = 67% to 70%
D = 63% to 67%
D minus = 60% to 63%
F = 0% to 60%
Boundary cases count as the higher of the two grades.

The syllabus, details and dates of each of the above assessments will be posted in Canvas.

Make-up exams and quizzes will be granted only for extenuating circumstances. Contact the instructor as soon as possible during the semester if you have such a circumstance. Absence from examinations and quizzes without prior approval will result in a score of 0.

Late Submissions Policy

Grades for late submissions are reduced by 10% per day they are late. For example, a homework is considered one day late if it submitted a couple minutes after the due date's 11:59 PM deadline. So, if a homework is submitted four-and one-half days late, then the max score it can receive is 50%. There is no point submitting a homework more than 9 days late as it already gets a zero then.

AI Use Policy

The concepts in this course form the foundation for a lifelong CS career, and mastering them requires direct engagement with the material. AI tools are permitted as a study aid: for self-quizzing, alternative explanations of concepts, and generation of additional practice problems. AI use is not permitted on any component of the assessed coursework tied to the learning objectives. To enforce this consistently, most assessment in this course takes place in class under uniform conditions.

Homework is not proctored, but AI use on homework is discouraged. Homework serves as the mechanism by which concepts are internalized: working through an algorithm, encountering difficulty, and resolving it independently is what builds the understanding required for quizzes, exams, and later professional application. Completing homework with AI assistance produces a completed assignment without producing the corresponding learning, and that gap typically surfaces during in-class assessment.

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

Here's an overview of the course, organized by lecture. The schedule is tentative and may evolve as the course progresses; any changes will be communicated in advance.

Week	Class Dates	Tentative Topic(s)
1	Aug 19	Introduction, Syllabus
2	Aug 24	Loop Invariants
2	Aug 26	Algorithm efficiency
3	Aug 31	Review (Queue, Linked List, Stack), Asymptotic Analysis
3	Sep 2	Divide and Conquer
4	Sep 7	No Class – Labor Day
4	Sep 9	Recursion Tree, Master Theorem
5	Sep 14	Heaps, Heapsort, Priority Queues
5	Sep 16	Quicksort + Quiz #1
6	Sep 21	Linear Time sorts
6	Sep 23	Order statistics, Quick Select
7	Sep 28	Binary Search Tree
7	Sep 30	Balanced Search Tree
8	Oct 5	RB Tree

Week	Class Dates	Tentative Topic(s)
8	Oct 7	Midterm #1
9	Oct 12	Hashing, Hash Tables
9	Oct 14	Dynamic Programming
10	Oct 19	Disjoint Sets
10	Oct 21	Graph Introduction, Graph Search
11	Oct 26	DAG, Topological Sort, Strongly Connected Components
11	Oct 28	Minimum Spanning Tree + Quiz #2
12	Nov 2	Prim's, Kruskal's Algorithms
12	Nov 4	Single Source Shortest Paths - Bellman Ford
13	Nov 9	Dijkstra's Algorithm
13	Nov 11	No Class – Veterans Day
14	Nov 16	All pairs shortest paths - Floyd Warshall
14	Nov 18	Midterm #2
15	Nov 23	NP-Completeness
15	Nov 25	No Class – Thanksgiving Break
16	Nov 30	NP-Completeness
16	Dec 2	NP-Completeness
17	Dec 7	Review for Finals
Finals	Dec 11	Final Exam