

COMP 110/L: Introduction to Algorithms and Programming

Spring 2026

Instructor Information

Instructor: Dr. Maryam Jalali
Email: mjalalita@csun.edu
Office: 4445
Office Hours: By appointment (Online via Zoom)

Course Web Page

<https://mjalalit.github.io/COMP110/>

Class Details

Lecture Sections

Section	Class Title	Days & Times	Room	Dates
COMP 110-07 (16125)	Lecture	TuTh 10:00–11:15 AM	JD 1600	Jan 20–May 8, 2026
COMP 110-08 (16227)	Lecture	TuTh 11:30 AM–12:45 PM	JD 1600	Jan 20–May 8, 2026

Lab Sections

Section	Class Title	Days & Times	Room	Dates
COMP 110L-07 (16126)	Lab	TuTh 2:00–3:15 PM	Online (Zoom)	Jan 20–May 8, 2026
COMP 110L-08 (16228)	Lab	TuTh 3:30–4:45 PM	Online (Zoom)	Jan 20–May 8, 2026
COMP 110L-10 (16612)	Lab	Fri 11:00 AM–1:45 PM	JD 1107	Jan 20–May 8, 2026

Important note for students in **COMP 110L-10 (16612) (in-person lab on Fridays)**: You may choose to attend one of the online lab sections on Tuesdays or Thursdays instead of the in-person Friday lab. If you attend an online lab during one of the Tuesday/Thursday time slots, you do not need to attend the in-person lab on Friday. Of course, if you prefer to attend the in-person lab on Friday, you are more than welcome to do so.

Zoom Link for Lab

Zoom link: <https://csun.zoom.us/j/84501556936>
Meeting ID: 845 0155 6936

Communication

Please use your CSUN student email for all communication. Keep emails short and focused and use a clear subject line beginning with **COMP 110 Question**. Always include your name, course, and CSUN email address. I generally respond within 24 hours on academic days.

Course Description (from the catalog)

Introduction to algorithms, their representation, design, structuring, analysis, and optimization. Implementation of algorithms as structured programs in a high-level language.

Goals

This course teaches basic skills in analyzing problems and solving them by finding or creating an appropriate algorithm (a recipe for solving a problem). Once an algorithm has been identified, it is translated into an executable application using the Java programming language. This last task is often called “programming,” although this step is only one among many important steps in the software development lifecycle. The principles and skills related to problem solving and algorithms are general and are not specific to any programming language.

The CSUN CS Department has made a policy decision to use the Java language for lower division courses such as COMP 110, so Java language details will be presented as an important part of both lecture and lab, although the policy may change in the future. For now, you must demonstrate an appropriate level of both problem solving and Java programming skill to successfully complete the course. The lecture component (COMP 110) focuses on concepts and practical examples. The lab (COMP 110L) focuses on developing working and correct applications using Java and problem-solving techniques covered in lecture.

This course does not assume any previous experience in computer programming and material begins at an introductory level. However, coverage is fast-paced and moves on to more advanced topics quickly. This is not a survey course for non-majors, it is a real programming course designed for students concentrating in Computer Science and related majors who need to quickly develop real programming skills.

Course Objectives

By the end of the course, students will be able to:

- Demonstrate the knowledge of a computer and operating system. The editing and compilation process.
- Translate human-readable algorithms represented by pseudo code, flowchart or flow block diagram to Java.
- Write and test Java programs using the 4 fundamentals of programming: sequence, choice, loop, and methods.

- Construct programs that require several methods and good knowledge of passing parameters.
- Demonstrate the knowledge of basic steps of software development: problem statement, program development, testing and documentation.
- Solve problems with one- and two-dimensional arrays.
- Use basic sorting and searching methods.
- Apply the class String.
- Read and write text files. Demonstrate practical using of Exception Handling.
- Recognize the role of Object-Oriented Programming in software development.
- Run the examples and exercises studied in the course.
- Understand the ideas of Polymorphism and Inheritance.

Prerequisites

Grade of “C” or better in MATH 102, 103, 104, 105, 150A or 255A, or a passing score on the Math Placement Test (MPT) that satisfies prerequisites for MATH 150A or 255A.

Corequisite

COMP 110L. Introduction to algorithms, their representation, design, structuring, analysis and optimization. Implementation of algorithms as structured programs in a high-level language. Lab: 3 hours per week. (Available for General Education, Lifelong Learning if required by student’s major.)

Course Materials

All course resources will be posted on: <https://mjalalit.github.io/COMP110/>

Videos, announcements, and grades will be available on Canvas: <https://canvas.csun.edu>

Any questions about a Lab/Exam grade should be addressed within two days of posting. After two days, all grades are final.

Textbook

No textbook is required. Optional reference:

- *Introduction to Java Programming and Data Structures*, Comprehensive Version, 11th Edition
- ISBN-13: 978-0-13-467094-2.

Any edition from the past several years will suffice.

Software Tools

This class requires you to have direct and continual access to a computer. In short, it is a de facto requirement to own your own laptop computer to be successful in your chosen field. You are free to do your work on your personal laptop. Lab machines use the Linux OS, but the software applications required for class work are available for all common OSes. You must install the Java Development Kit (JDK) and an editor to write programs for the class. You can use whatever editor you prefer. If you don't have a preference, then I suggest JGrasp. It is important that you install the JDK first, then the JGrasp editor. Make sure you install the JDK, not the JRE (Java Runtime Environment).

For the JDK, go to:

- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Download the Java SE (Standard Edition) JDK, the latest version. Install JDK first, before proceeding to download and install JGrasp.

For JGrasp, go to:

- <http://www.jgrasp.org>
- Click on the Downloads link and download the installation for your platform (Windows, MacOS, or Linux/Unix).
- This software is available in JD1104/05/07 and in the open computer science labs.

Course Format and Required Technology Resources

To succeed in this course, you must have reliable access to a computer and internet connection. CSUN offers currently enrolled students the option to borrow devices such as computers and internet hotspots through its Device Loaner Program.

Course Etiquette

All lab sessions will be conducted online via Zoom. Attendance is strongly encouraged, as active participation is essential for completing lab activities and benefiting from real-time support. You are not required to use video during lab; however, your full name must be visible as your screen name when joining Zoom. Students will be muted upon entry to minimize disruptions and should remain muted unless instructed otherwise. Zoom features such as chat, raise hand, and breakout rooms will be used during lab sessions. Please contact me if you have any questions or special requests regarding the lab format or policies.

Accessibility, Academic, and Other Support Resources for Students

Information about accessibility, academic, and other student support resources is available at this link.

Grading

You will receive a single combined grade for the lecture and lab. Your grade is based on the following components. **Canvas will be only used for posting the grades of the exams, quizzes and labs. At any point of semester, if you want to estimate your performance in the class, you must use the following table.**

Lecture Participation	5%
Lab Assignments	20%
Quizzes	5%
Lab Exam 1	10%
Lecture Exam 1	15%
Lab Exam 2	15%
Lecture Exam 2	30%

Lab assignments will be frequent, typically with two due per week. The exact number of lab assignments has not been set, as this will depend somewhat on how the class progresses. These are low-stakes assignments which are intended to introduce an entirely new concept for the first time. Lab assignments are submitted through Canvas (<https://canvas.csun.edu/>). In the event that there is a problem with Canvas, you may email your assignment to me (mjalalita@csun.edu), though this should be considered a last resort.

Grading Policy

Exam questions will be based on both the lecture slides and the material discussed in class. To succeed, you should attend class regularly, participate in discussions, complete all assignments, and take notes. If you miss a class, be sure to review the posted materials and go over the key points covered. **I regularly conduct in-class practice activities, including live coding exercises, which will not be posted on Canvas or the course website. If you miss a class, it is your responsibility to ask a classmate about the in-class content. Any material discussed during class may appear on the exams.**

Minimum Score	Grade
92.5	A
89.5	A-
86.5	B+
82.5	B
79.5	B-
76.5	C+
72.5	C
69.5	C-
66.5	D+
62.5	D
59.5	D-
0	F

- Plus/minus grading is used, according to the scale below. The left column shows the minimal score necessary to receive the grade in the right column. The highest letter grade possible given the score is chosen; e.g., if you receive an 88.2, you'd receive a 'B+' for the course, which corresponds to being ≥ 86.5 .
- An incomplete (I) grade is given for genuine medical and other certified emergencies only; it is never given to catch up with missed assignments. Furthermore, to receive an Incomplete grade, you must have successfully completed at least two-thirds of the semester with a passing grade.
- **Make-up exams must be arranged prior to the scheduled date and will be allowed only at the discretion of the instructor. There will be no make-up quizzes.**

Plagiarism and Academic Honesty:

Plagiarism in any assignment or cheating in the examinations will result in a grade of F in the entire course. On an exam, you are expected to submit only your own work. On a programming project, it is permissible to discuss solution approaches in a general sense with other students. But when submitting a program for a grade, the program must represent your own work. It cannot be a copy of another student's program, even if you worked in a group with that student. **Penalties for academic dishonesty on a single exam or programming project may result in a grade of "F" for the entire course. A report will also be made to the Office of the Vice President for Student Affairs.** Students who repeatedly violate this policy across multiple courses may be suspended or even expelled. If you have any doubts about what is considered dishonest, please ask the instructors for guidance before taking such a serious risk. In general, full disclosure is the best policy on any submission. In other words, if a friend helped you to complete a project, state this fact in writing at the beginning of the submission. Such submission may not earn full points.

Plagiarism or cheating results in a grade of F for the course and a report to the Office of the Vice President for Student Affairs. All submitted work must be your own.

Late Policy

Late assignments will be accepted without penalty if prior arrangements have been made or there is some sort of legitimate emergency (at my discretion). If you must be absent from an exam, contact me ASAP to see if alternative accommodations can be made. If an assignment is otherwise submitted late, it will be penalized according to the following scale:

Days Late	Penalty
1	10%
2	30%
3	60%
4+	No Credit

To be clear, assignments which are submitted four or more days beyond the deadline will not receive credit.

Attendance

Attendance will not be strictly monitored; however, active participation is an important component of this course and accounts for 5% of the final grade. Scheduled lecture and lab times are the primary opportunities to ask questions, participate in discussions, and receive help on course material and projects. Students are expected to attend lecture and lab regularly and engage in class activities. I am not able to provide extensive explanations of class material outside of scheduled class time (including excessive email) for students who do not attend lecture or lab.

Disability Resources

If you have a disability and need accommodations, please register with the Disability Resources and Educational Services (DRES) office or the National Center on Deafness (NCOD). The DRES office is located in Bayramian Hall, room 110 and can be reached at (818) 677-2684. NCOD is located on Bertrand Street in Jeanne Chisholm Hall and can be reached at (818) 677-2611. If you would like to discuss your need for accommodations with me, please contact me to set up an appointment.

Changes to Syllabus

Changes may be needed to this syllabus and to the course plan. All such changes will be announced in class and will be announced via Canvas. Students are responsible for this information.

List of Topics (subject to change)

Week	Topic
1	Compiling and running Java code
2	User input and arithmetic
3	Methods and testing
4	Objects and instance variables
5	Overloading and conditionals
6	Switch and complex conditionals
7	Arrays and loops
8	Advanced loops and arrays
9	Complex loops
10	Arrays of objects
11	Strings and multidimensional arrays
12	Inheritance and polymorphism
13	Interfaces and exceptions
14	File I/O and finally