COMP 110: Intro to Algorithms and Programming

Instructor: Bahram Zartoshty (bzartoshty@csun.edu)
Course Web Page: https://bzartoshty.github.io/

Class Title	Class Title	Days & Times	Room	Class Date
COMP 110-01 (10306)	INTRO ALGRTH/PROG (Lecture)	MoTuWeTh 8:00AM - 9:35AM	Online Class	May 27, 2025- Jul 8, 2025
COMP 110L-01 (10308)	ALGRTH/PROG LAB (Laboratory)	MoTuWeTh 9:45AM - 11:20AM	Online Class	May 27, 2025- Jul 8, 2025

Final Exam: July 8, 2025

Zoom Information:

Zoom id: 8186772625
Zoom Password: JD4441

Office Hours:

Location: **ZOOM**
Office Hours: By Appointment

Communication:
Please use email for messages. Please keep emails short and focused, and use a clear subject line beginning with "COMP 110 Question". Many technical questions are better handled during lectures and lab sessions than by email since the class as a whole might benefit from the discussion.
You may email me (bzartoshty@csun.edu) at any time; I will generally respond within 24 hours (during the academic days). Always include your name, course, and CSUN e-mail address in your messages. An email address like meqwik@love.com leaves me clueless about who you are**!**

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: A successful student will be able to:
1. Demonstrate the knowledge of a computer and operating system. The editing and compilation process.
2. Translate human-readable algorithms represented by pseudo code, flowchart or flow block diagram to Java.
3. Write and test Java programs using the 4 fundamentals of programming: sequence, choice, loop, and methods.
4. Construct programs that require several methods and good knowledge of passing parameters.
5. Demonstrate the knowledge of basic steps of software development: problem statement, program development, testing and documentation.
6. Solve problems with one- and two-dimensional arrays.
7. Use basic sorting and searching methods.
8. Apply the class String.
9. Read and write text files. Demonstrate practical using of Exception Handling.
10. Recognize the role of Object-Oriented Programming in software development.
11. Run the examples and exercises studied in the course.
12. 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 Material:

- Most of the resources (Slides, codes) will be posted on : **https://bzartoshty.github.io/COMP110/**
- Videos and announcements will be available on Canvas (https://canvas.csun.edu).
- Grades will be posted 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:

There is a ***required*** interactive e-textbook:

It's time to access Revel for Liang, Introduction to Java Programming and Data Structures, 13e, the course materials for COMP 110/L Introduction to Algorithms.

When you're ready, join your course at:

Use the "Access Pearson" on CANVAS
Or
Under Modules → Revel assignments → click
1: Introduction to Computers, Programs, and Java

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 OSs. 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. You must 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).

Course Format and Required Technology Resources:

This course will be taught entirely online (no campus meetings will be required), in a synchronous format (this course will have scheduled live meeting times). All activities, assignments, and exams in this course will be completed via Canvas. 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.

Exams:

Exams, programming assignments ("labs"), and projects are based on what is presented in the lecture. There will be a Midterm and a Final Exam. All these exams will have one part that is multiple choice. The Midterm and Final Exams will have an additional section for writing *code segments* to demonstrate capability.

Lecture:

All lectures will be delivered via instructor PowerPoint slides, coupled with Revel material and coding examples.

For the best level of preparation, students should attend every lecture and lab and participate in discussions, rather than simply reading the lecture or lab material on their own. Use of the Revel will provide a means of learning by *interaction* – assignments will be graded.

Lab: Assignments

All Labs and Projects are graded programming assignments.

Labs: There will be lab assignments for each chapter of Revel Book. They are carefully selected to integrate the current lecture material and lead the student to mastery of traditional and fundamental programming techniques, areas, and algorithms.

- Revel Labs will be issued, performed, and auto-graded in appropriate chapters inside Revel, guiding students through the recommended program structure.
- IDE Labs are designed to fully incorporate all the material learned in the term of the course. They are opportunities for the student to demonstrate their mastery of this course. Projects will be done using an "Integrated Development Environment (IDE)".

Submissions of "IDE Labs" will use a provided form for documentation to include each student's Requirements, Input, Output, and Source Code listing. "IDE Labs" are to be completed on their dedicated forms and submitted as Word files to CANVAS. Source code (.java) should also be uploaded as part of the submission.

Lab: Programming – Integrated Development Environment (IDE)

Programming in Java is practiced in each Lab session, putting to use concepts and constructs learned in the preceding lectures. We will use the Java *Software Development Kit* (SDK) called ***JDK***, from Oracle, for the Java *compiler*. We will use the Revel IDE (when using the Revel) and ***jGRASP IDE*** (freeware from Auburn University – www.jgrasp.org).

Learning to use the jGRASP IDE is an important part of the lab and the course. All final source code editing, compiling, running, and debugging will be done with the IDEs. You will be shown how to utilize some of its extra features, such as generating *CSD* (Code Structure Diagram) and *UML* (Universal Modeling Language) class diagrams, along with running programs in *debug mode*.

The Lab sessions will mostly involve the instructor going over the assignments, along with the "starter code" for each. Some related general code segments, and some other example code, may also be presented during the labs. During the Lab sessions, students are encouraged to work on their assignments and may ask for help from the instructor or the in-class tutors, or any other students.

It will usually be necessary to continue to work on your programming assignments on your own time outside of your official lab time. You will also need to work on your Projects in parallel with Lab programs as part of your lab work.

Collaboration

Collaboration among students is encouraged. However, this must fall short of copying any code segments, which is plagiarism. (That said, it is often the case that there is only one good solution to each part of an assignment, and this instructor will take that into account.)

Late Submission Policy:

All assignments and labs will have a due date. No assignments will be accepted after the due date.

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.

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.

Category	Weight
Revel Assignments	30%
Lab Assignments	20%
Midterm Exam	25%

Final Exam	25%

Lab assignments will be frequent, typically with 2-3 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 (bzartoshty@csun.edu), though this should be considered a last resort. **No Late lab assignments will be accepted.**

Grading Policy: Exam questions will relate to the contents of both the textbooks **and** *material discussed in class*. To do well, you should attend virtual classes regularly, participate in discussions, do all assignments, and take notes. If you miss a class, please make sure to review the posted material, including the videos for the lecture, and go over the important points with them.
Exams are closed book/closed note
The total category grade is calculated based on the sum of all grades for the category.

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.

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.

If your score is >=...	**...you will receive...**
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

NOTE: Failure to take the Final Exam will result in a grade of "WU" which is equivalent to a grade of "F"
An important part of this course is the notation, terminology, concepts, and definitions; therefore, I do not answer questions during examinations.

In fairness to all, I don't give make-up for any missed projects, quizzes, or exams.
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.

Plagiarism and Academic Integrity:

Academic integrity is a core value of the academic community. It is essential for maintaining the quality of higher education, for the development of critical thinking skills, and for ensuring that academic achievements are based on merit. College of the Canyons defines academic integrity as the ethical and honest pursuit of knowledge, scholarship, and intellectual growth. It involves upholding the values and principles that guide ethical behavior in academic work, including honesty, fairness, trust, and respect for the intellectual work of others.

At CSUN, we believe that academic integrity is one of the most important qualities college students need to develop and maintain. Conversely, academic dishonesty is any practice or behavior, whether intentional or unintentional, that undermines the integrity of material submitted for academic credit. Academic dishonesty violates the principles of academic integrity and can have serious consequences for both the individual and the academic community. Common types of academic dishonesty fall into one of three categories:

1. Plagiarism or self-plagiarism.

a. Plagiarism is submitting someone else's work as one's own, or without adequate or proper attribution, including unauthorized use of AI-generated material.
b. Self-plagiarism is resubmitting the same work from a different class without the current instructor's knowledge and approval (e.g., using the same programming exercise solution in more than one class).

2. Cheating. Examples of cheating include (but are not limited to):

a. Copying from someone else's quiz or exam.
b. Using an unauthorized aid during an exam (e.g., phone, smart watch, notes or browser).
c. Unauthorized collaboration or communication with others, over services like Discord, during a quiz or exam, including downloading, uploading, sharing or duplicating course material.
d. Unauthorized use of generative AI like ChatGPT or other web-based applications like stackoverflow.com, chegg.com, coursehero.com, Quizlet during a quiz or exam.

3. **Fraud.** Examples of fraud include (but are not limited to):

a. Having a quiz or exam completed by someone else.
b. Buying, selling or otherwise obtaining or quizzes or exams.
c. Falsifying, misrepresenting or forging an academic record or supporting documents (e.g., submitting a fake doctor's note, misrepresentation of identity).

d. Improper access/obstruction of materials/systems (e.g., stealing a quiz or exam).
e. Falsely claiming participation in a breakout room.

It is the responsibility of all members of the academic community to uphold the principles of academic integrity and to prevent instances of academic dishonesty. Cases of alleged academic dishonesty may be referred to Student Conduct for investigation. Relevant disciplinary action policies can be found in the Student Conduct Code.

Accessibility, Academic, and Other Support Resources for Students:

https://docs.google.com/document/d/1MFULw4Ybpyu3FliHUClzjxmsHxCGSBFRApNxycQTCHw/edit?usp=sharing

Disabled Students:

"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."

Attendance:
Attendance will not be monitored as closely, but note that the scheduled lecture and lab time is the most appropriate time to ask questions and get help on your projects. I am not willing to spend an unreasonable amount of time outside of class (including excessive use of email) to explain class material to anyone who is not attending lecture and lab. In other words, you are expected to participate in lecture discussions by attending lecture and get programming help on your projects in person by attending lab.

Changes to Syllabus:

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

List of Topics (subject to change)
Exactly which topics are covered and when is subject to change.

Week #	Topics
1	**Introduction to Computers, Programs, and Java**
2	**Elementary Programming**
3	**Selections**
4	**Mathematical Functions, Characters**
5	**Loops**
6	**Loops continue**

7	**Single-Dimensional Arrays**
8	**Multi-Dimensional Arrays**
9	**Methods, searching, sorting**
10	**Objects and Classes**
11	**Thinking in Objects**
12	**Inheritance and Polymorphism**
13	**Exception Handling**
14	**Text/IO (time allowed)**
15	**Interfaces (time allowed)**