COMP 110/L: Intro. to Algorithms and Programming
Fall 2022

Instructor: Kyle Dewey (kyle.dewey@csun.edu)
Course Web Page: https://kyledewey.github.io/comp110-fall22
Office: JD 4419

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.

Learning Objectives

A successful student will be able to:

- Write programs which perform input / output, through the console and files
- Write programs which use variables, methods, conditionals, loops, arrays, classes, objects, and exceptions
- Run tests, write tests, and debug their code when tests fail

Java is used throughout the class, though these skills are broadly applicable to most programming languages.

Textbook

No textbook is required. If you'd like a textbook for further study, a good supplemental textbook is Daniel Liang's "Introduction to Java Programming". Any edition from the past several years will suffice.

Grading

You will receive a **single combined grade** for the lecture and lab. Your grade is based on the following components:

Lab Assignments	30%
Lab Midterm Exam #1	8%
Lab Midterm Exam #2	15%
Handwritten Midterm Exam	15%
Lab Final Exam	16%
Handwritten Final Exam	16%

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 (kyle.dewey@csun.edu), though this should be considered a last resort.

Plus/minus grading is used, according to the scale below:

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

Why so many lab assignments?

Software development is analogous to swimming. You cannot learn to swim just by reading a book or listening to a lecture. Instead, you must spend hours in the pool on an incremental basis. While you may be able to "cram" random trivia, you cannot cram the capability to swim. Similarly, you cannot cram software development: you must practice it frequently. These assignments give you plenty of opportunities to practice.

Collaboration for Lab Assignments

All students are required to submit their own individual work. For lab assignments (and **only** lab assignments), students may discuss among each other, as long as they don't simply copy/paste each other's solutions. However, you **may** discuss your actual code with someone else, including viewing the code on a monitor. The only stipulation is that **if you do discuss with someone else, say so in your submission.** This is not for punitive reasons; this is only so I get a sense of who is working with who. My intention with this policy is to enable collaborative learning, as opposed to simply sharing a solution. Additionally, keep in mind that in an exam setting, you'll have to do everything alone, so you should be prepared to do so.

Plagiarism and Academic Honesty

While collaboration is allowed on lab assignments, you are responsible for all of your own work. You may **not** take code from online sources and submit it as your own. If you must take code from online, cite where you took the code from. Worst-case scenario, you'll receive a 0 for whatever you took, but no further action will be taken. In general, code taken online which solves more general things (e.g., "how do I iterate through an array in Java") is more acceptable than code which solves more specific things (e.g., "how do I implement a recursive find function over immutable linked lists in Swift"). General bits of code only give you pieces of a solution, whereas specific bits of code often will give you a complete copy/pastable solution. If it's not 100% clear if something is permitted to be used or not, you can always ask me beforehand.

Chegg is specifically disallowed as an online resource, as it's almost always used as a repository of complete questions with answers. That is, the questions/answers are practically always of the specific kind mentioned above, with zero learning whatsoever.

No discussion is allowed during exams, except with the instructor. Any violations can result in a failing grade for the assignment/exam, or potentially failing the course for egregious cases. A report will also be made to the Dean of Academic Affairs. Students who repeatedly violate this policy across multiple courses may be suspended or even expelled.

Attendance

In the first week of class, I will take attendance. If you miss both sessions in the first week and have not made alternative arrangements with me, you must drop the class, as per University policy (http://catalog.csun.edu/policies/attendance-class-attendance/). After the first week I will not take attendance, and attendance is not mandatory, though you are strongly encouraged to attend. I enforce this policy in order to help students on the waitlist get into the course.

Communication

In general, any questions should be made through Canvas. You can also email me, though I'm usually much faster to respond to Canvas than my general email.

Late Policy / Exam Scheduling

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:

If your assignment is late by <= this many days...	...it will be deducted by...
1	10%
2	30%
3	60%
4+	100%

To be clear, assignments which are submitted four or more days beyond the deadline will not receive credit. The reason for such a harsh late policy is that we will generally discuss solutions in class shortly after the deadline, and this late policy discourages people from simply pulling a solution from an in-class discussion.

Class Feedback

I am open to any questions / comments / concerns / complaints you have about the class. If there is something relevant you want covered, I can push to make this happen. I operate off of your feedback, and no feedback tells me “everything is ok”.

---**Class Schedule and List of Topics on the next page**---

Class Schedule and List of Topics

Exactly which topics are covered and when is subject to change. The dates of the exams are fixed, and will not be changed.

Week	Dates	Topics
1	8/29 8/31	Compiling and running Java code
2	~~9/5~~ 9/7	Labor day 9/5 (no class); reading user input
3	9/12 9/14	Arithmetic operations; methods
4	9/19 9/21	Software testing; review **9/21 Lab Midterm Exam #1**
5	9/26 9/28	Introduction to objects, instance variables, visibility modifiers, overloading
6	10/3 10/5	basic conditionals, random numbers, floating point
7	10/10 10/12	complex conditionals, `switch`, more random numbers, command-line arguments, arrays
8	10/17 10/19	Loops, more arrays
9	10/24 10/26	Complex loops, `String.split`, multidimensional arrays
10	10/31 11/2	More multidimensional arrays, review
11	11/7 11/9	**11/7 Lab Midterm Exam #2,** **11/9 Handwritten Midterm Exam**
12	11/14 11/16	Inheritance, `super`, polymorphism
13	11/21 11/23	Interfaces, exceptions
14	11/28 11/30	File I/O, `finally`
15	12/5 12/7	Spillover / review
16	12/12 12/14	**12/12 Lab Final Exam** **For 9 AM section: Lecture Final Exam 8-10 AM, 12/14, in JD 1104** **For 12 PM section: Lecture Final Exam 12:45 PM - 2:45 PM, 12/19, in JD 1104**