Computer Science II

CSCE 156 - Syllabus - Spring 2025

School of Computing
College of Engineering
University of Nebraska-Lincoln
University of Nebraska-Omaha

> "Smart data structures and dumb code works a lot better than the other way around."
>
> —Eric S. Raymond, *The Cathedral and the Bazaar*

Course Info

Prerequisites: MATH 106 or equivalent and a grade of C or better in CSCE 155

Description: Data structures, including linked lists, stacks, queues, and trees; algorithms, including searching, sorting, and recursion; programming language topics, including object-oriented programming; pointers, references, and memory management; design and implementation of a multilayer application with SQL database.

Credit Hours: 4

Postrequisites: This course is a prerequisite for CSCE 310 - Data Structures & Algorithms. A grade of C or better is required to move on to CSCE 310.

University Policies, Resources & Services

Students are responsible for knowing the university policies and resources found on this page: https://go.unl.edu/coursepolicies

Learning & Skills Objectives

Throughout the duration of this course, you will be presented with a lot of information, some of which will have immediate application and some whose importance is not immediately apparent. As a different measure of your success in this course, by the end, you should have the following skills.

- Be able to more easily pick up and learn a new programming language on your own

- Be able to recognize, use, design and appreciate programs with an object-oriented flavor

- Be able to design a database-backed application from scratch given a general problem statement

- Be able to understand and implement basic data structures as well as recognize how and when to leverage appropriate simple data structures

- Be able to identify and analyze basic algorithms

Overall, CS2 is a continuation of CS1. At the end of a typical CS1 course, you should be able to approach a reasonably complex problem, design a top-down solution, and code a program in a high-level programming language. By the end of this course your approach to problem solving should be modified to ask and answer design questions first:

- What entities/objects would be appropriate to model this problem?

- How would these objects interact to solve the problem?

- What data structures would be the most appropriate or efficient to use?

Tentative Topics

- Introduction to Java / Python (Honors section)

- Object-Oriented Programming, OOP in Java

- Database Design, SQL, Database Connectivity, multi-tier design

- Lists (array-based, linked; circular doubly-linked), Stacks & Queues

- Algorithm Analysis, Recursion, Searching & Sorting

- Trees; Binary Search Trees

Continuation of Instruction

If in-person classes are canceled by the University during a lecture day, lecture may still be livestreamed.

Assessment

Assessment (grading) will be based on labs and assignments. To start you out on the right foot, you have been awarded 20 free starter points. You are currently earning an A in this course; keep it up!

Category	Number	Points
Starter Pts		20
Surveys		30
Labs	14	350
Assignments	8	950
Total		1350

The honor section point distribution is similar but there are more total points due to additional required points on the assignments.

Scale

Final letter grades will be awarded based on the following standard scale. This scale may be adjusted upwards if the instructor deems it necessary based on the final grades only. No scale will be made for individual assignments.

Letter Grade	Percent
A+	>= 97%
A	>= 93%
A-	>= 90%
B+	>= 87%
B	>= 83%
B-	>= 80%
C+	>= 77%
C	>= 73%
C-	>= 70%
D+	>= 67%
D	>= 63%
D-	>= 60%
F	< 60%

Labs & Attendance

There are 14 labs. They are graded based on completion which means *all* of the test cases must compile, run, **and pass**. If any test(s) fail no points will be awarded. Each lab is worth 25 points: 20 for the actual lab and 5 points for attendance at lab. However, if you cannot attend or choose not to attend you can still earn the 5 points back by earning full credit on the lab itself.

Labs give you hands-on exercises for topics recently covered in lecture. The purpose of lab is not only to give you further working experience with lecture topics, but also to provide you with additional information and details not necessarily covered in lecture. Each lab will come with some starter code and exercises that you have to complete. You will submit your code as specified through the handin and we will grade your lab using the webgrader.

Labs are setup as a *peer programming* experience. If you are attending lab *in person*, you may be randomly paired with a partner. One of you will be the *driver* and the other will be the *navigator*. The navigator will be responsible for reading the instructions and guiding the driver. The driver will be in charge of the keyboard and will type the code. Both driver and navigator are responsible for developing and working through solutions *together*. Neither the navigator nor the driver is "in charge." This is an equal collaboration and all efforts should

be equally shared. Beyond your immediate pairing, you are encouraged to help and interact and with other pairs in the lab. If you are attending lab remotely or online, you may be paired up with another person virtually if you choose or you may complete the lab on your own.

Assignments

The majority of your assessment will be based on assignments whose point values may vary. There are 8 assignments as well as a design document. Assignments will be due 1 second before midnight Friday evening the week that they are due. Code and other relevant *artifacts* must be submitted using handin unless otherwise stated.

In this course, most of the assignments build on prior assignments. Therefore, it is **extremely important** for you to stay on task and not fall behind with assignments. You should start on assignments early and resolve any and all issues (bugs, design flaws) as soon as possible so they do not carry over to the next assignment. To facilitate good time management, we have provided a **Gantt Chart** with a recommended schedule of work.

15th Week Policy Notification

A per UNL's 15th Week Policy (available here: https://registrar.unl.edu/acad emic-standards/policies/fifteenth-week-policy/) we are required to serve written notice that the final assignment as well as the final lab will be due during the 15th week or "dead week."

Grading Policy

If you have questions about grading or believe that points were deducted unfairly, you must first address the issue with the individual who graded it to see if it can be resolved. Such questions should be made within a reasonable amount of time after the graded assignment has been returned. No further consideration will be given to any assignment a week after it grades have been posted. It is important to emphasize that the goal of grading is consistency. A grade on any given assignment, even if it is low for the entire class, should not matter that much. Rather, students who do comparable work should receive comparable grades (see the subsection on the scale used for this course).

Late Work Policy

In general, there will be no make-up or late work accepted. Exceptions may be made in certain circumstances such as health or emergency, but you must make every effort to get prior permission. Documentation may also be required.

Homework assignments have a strict due date/time as defined by the server's system clock. All program files must be handed in using handin as specified in individual assignment handouts. Programs that are even a few seconds past

the due date/time will be considered late and you will be locked out of handing anything in after that time.

Webgrader Policy

Failure to adhere to the requirements of an assignment in such a manner that makes it impossible to grade your program via the webgrader means that a disproportionate amount of time would be spent evaluating your assignment. For this reason, we will not grade any assignment that does not compile and run through the webgrader.

Academic Integrity

All homework assignments, programs, and exams must represent your own work unless otherwise stated. No collaboration with fellow students, past or current, is allowed unless otherwise permitted on specific assignments or problems. The School of Computing has an Academic Integrity Policy. All students enrolled in any School of Computing course are bound by this policy. You are expected to read, understand, and follow this policy. Violations will be dealt with on a case by case basis and may result in a failing assignment or a failing grade for the course itself.

Use of AI   The use of any AI tools for the design or implementation of programs is strictly prohibited for this course.

Honors Section

For those in the Honors section of this course topics will be covered at a faster pace so that additional "advanced" topics can also be covered.

In addition, each assignment may contain additional requirements or problems that must be completed. There may also be additional assignments. Bottom line, there will be higher expectations with regard to the quality and quantity of work in this course compared to the regular section. As a student in this course, you will be expected to meet or exceed those expectations which may require more of your time and a more proactive approach to studying and greater diligence in completely assignments.

Communication & Getting Help

The primary means of communication for this course is Piazza, an online forum system designed for college courses. We have established a Piazza group for this course and you should have received an invitation to join. If you have not, contact the instructor immediately. With Piazza you can ask questions anonymously, remain anonymous to your classmates, or choose to be identified. Using this open forum system the entire class benefits from the instructor and TA responses. In addition, you and other students can also answer each other's

questions (again you may choose to remain anonymous or identify yourself to the instructors or everyone). You may still email the instructor or TAs, but more than likely you will be redirected to Piazza for help.

In addition, there are is an anonymous suggestion box that you may use to voice your concerns about any problems in the course if you do not wish to be identified.

Learning Assistant Program (LAP)

This course is supported by the CSE Learning Assistant Program (LAP). The mission of the LAP is to improve student comprehension and retention in computing fields by focusing on the learner's experience. This course will be supplemented by Learning Assistants (LAs) and Course Leaders (CLs) to help improve your learning. LAs and CLs are other undergraduate students who have taken the same or similar courses and have been trained to help you succeed in this course. Your LAs and CLs will hold regular office hours, help with grading, and assist you with labs/assignments.

I strongly encourage you to utilize the LAs and CLs when you are completing coursework. More information can be found in the Learning Assistant Program Module on Canvas.

UNL Writing Center

The Writing Center can provide you with meaningful support as you write for this class as well as other courses. Trained undergraduate and graduate peer consultants are available to talk with you about all forms of communication. You are welcome to bring in everything from lab reports, presentations, and research papers to cover letters, application essays, and graduate theses and dissertations. Writing Center Consultants can work with you at any stage of the writing process, from brainstorming and organizing your ideas through polishing a final draft.

There are three ways you can connect with a Consultant:

- Online (a real-time, video conversation)
- eTutoring (email feedback)
- In person in Andrews 102 To learn more about online options and view video tutorials, visit https://www.unl.edu/writing/online-writing-center-services.

Sign up for an appointment any time by visiting https://unl.mywconline.com. For other information about the Writing Center, please visit https://unl.edu/writing.

Getting Help

Your success in this course is ultimately your responsibility. Your success in this course depends on how well you utilize the opportunities and resources that we

provide. There are numerous outlets for learning the material and getting help in this course:

- Lectures: attend lectures regularly and when you do use the time appropriately. Do not distract yourself with social media or other time wasters. Actively take notes (electronic or hand written). It is well-documented that good note taking directly leads to understanding and retention of concepts.

- Lecture Videos: Lecture videos are intended as a supplement that mirrors lecture material but that may not cover everything. Watch them at your own pace on a regular basis for reiteration or in case you missed something in lecture.

- Required Reading: do the required reading on a regular basis. The readings provide additional details and depth that you may not necessarily get directly in lecture.

- Labs: use your time during labs wisely. Engage with your lab instructors, teaching assistants, your partner(s) and other students. Be sure to adequately prepare for labs by reading the handouts before coming to lab. Get started and don't get distracted.

- Piazza: if you have questions ask them on Piazza. It is the best and likely fastest way to get help with your questions. Also, be sure to read other student's posts and questions and feel free to answer yourself!

- Office Hours & Student Resource Center: the instructor and teaching assistants hold regular office hours throughout the week as posted on the course website. Attend office hours if you have questions or want to review material. The Student Resource Center is open Monday through Friday. Even if your TAs are not scheduled during that time, there are plenty of other TAs and students present that may be able to help. And, you may be able to help others!

- Don't procrastinate. The biggest reason students fail this course is because they do not give themselves enough opportunities to learn the material. Don't wait to the last minute to start your assignments. Many people wait to the last minute and flood the TAs and SRC, making it difficult to get help as the due date approaches. Don't underestimate how much time your assignment(s) will take and don't wait to the week before hand to get started. Ideally, you should be working on the problems as we are covering them.

- Get help in the *right way*: when you go to the instructor or TA for help, you must demonstrate that you have put forth a good faith effort toward understanding the material. Asking questions that clearly indicate you have failed to read the required material, have not been attending lecture, etc. is *not acceptable*. Don't ask generic questions like "I'm lost, I don't know what I'm doing". Instead, explain what you have tried so far. Explain

why you think what you have tried doesn't seem to be working. Then the TA will have an easier time to help you identify misconceptions or problems. This is known as "Rubber Duck Debugging" where in if you try to explain a problem to someone (or, lacking a live person, a rubber duck), then you can usually identify the problem yourself. Or, at the very least, get some insight as to what might be wrong.