

Quick Facts

What does this course offer?

This class continues the first data structures class, building on lists, recursion, and sorting to tackle advanced structures like symbol tables and graphs. We will get hands-on experience both using and building these data structures and a chance to expand your knowledge even further in an area of your interest. You will see the concepts in theory and practice.

Success in this class looks different than many others. You will need to learn new ideas and skills, but real success comes in applying those skills in creative ways. The best path to this is your deliberate practice within various skills. You must become resilient to failure and always learn from your mistakes. Programmers make and fix mistakes all day long to claw towards useful solutions. More than anything, if you can build this mindset – learning through failure rather than avoiding it – you will be well-positioned to be a potent computing professional.

Where do I find more details?

D2L will host most course information, including assignment details, grades, and content. Unlike many courses, you have many choices, which you will express in your course plan. This course uses a concept known as a *flipped classroom*. I will not lecture you each week, give you a watered-down set of homework, and hope you remember enough on the test. Instead, you should read the book, watch course videos, and start programming and use class time to work through your challenges rather than wait until class to get started. We will tackle content in class, but starting at the middle rather than the beginning.

Our scheduled meeting times include:

Class - W 5:45 - 9:00 PM CST

We start on Zoom for the winter quarter (see link on D2L) and return to the classroom as policy dictates. Zoom will always be an acceptable way to attend or watch afterward. When in the classroom, you must adhere to all requirements put forward by the University and teaching team.

In addition, to Zoom, we will use Discord to collaborate, particularly outside of class times. Look for a link to sign on to the Discord server in D2L, and make sure to stay in contact with your peers and instructional team.

How will I be graded?

This class has a very different grading approach than previously encountered. Many classes grade you on your “percentage correct”, with few opportunities to recover from mistakes. I prefer a system that rewards effort and results without arbitrary time restrictions. Nearly every task earns points, and you continue to earn points until you achieve your desired grade (and show minimal competency across each topic for top grades). There are hundreds more points available than required, even for a top grade, so you can choose challenges of interest to you rather than decided by me. You determine what grade you earn and how you earn it within the tasks offered.

People who do best in this course dedicate time and make some progress each week or make up the work when they fail to do so. I give no high-stakes tests, and you can resubmit most assignments until the end of the course. Programmers make mistakes every day; it is a natural part of the process! I expect you to learn to correct your errors (seeking the appropriate help when needed) and resubmit the work until ‘perfect’. However, this means you need to put in the time to work through the tough parts.

The tasks that have due dates revolve around professional communications. Each week you can earn points by completing a status report. You can also earn points by maintaining a Course Plan indicating your goals and progress. While I want to give you choices when you complete your work, flexibility comes with planning and communicating your plan – a good habit for the professional world. If you know you tend to be a procrastinator and work better under traditional deadlines, see the Personal Accountability Contract for support (under Professionalism in D2L).

Learning aids

The primary textbook is *Algorithms 4e* by Robert Sedgewick and Kevin Wayne (Addison-Wesley, 2011). We will use a narrow slice of this book for this course, the same text from the prior class, but sadly not available for free at the library any longer. It is known as an ebook [here](#).

My Philosophy of Learning

My research investigates how people think and learn, and as such, I try to embody these ideas in my classes. I believe that given the right circumstance and enough time, anyone can learn what they seek. To that end, I have some very different ideas about organizing a formal classroom. My approach to the work and grading will likely be very different from your past experiences. I will ask you to repeatedly put your knowledge into action and work through the 'hard parts' until you know your stuff.

One of the most important things I hope to share is techniques for independently learning complex technical topics. Successful professionals take ownership of their learning rather than waiting for someone to tell them what and how to do their work. Computing moves too fast to wait for someone to create a class! Each person brings strengths and has gaps in their experience based on your life's journey.

I hope to allow you to leverage those strengths and let you choose the speed and depth at which you tackle the course's content. If you need time to understand a topic better, I won't punish you with arbitrary late penalties. If you are highly motivated, you can dive in deeper and take on greater challenges. Beyond learning a specific topic, I hope to show you what it is to be both a competent technologist and a working professional.

Competence versus Professionalism

Professionals "know their stuff" and how to work with and be accountable to others. Your future employer will evaluate you on both your competence and professional conduct. Developing good communication and time management habits are critical to your academic and professional success, and this class asks you to start building those habits of mind.

This class asks you to submit a weekly status report (due by every Sunday!) and manage a plan for completing the course, among other professionalism tasks. Your plan will guide your coursework focus your learning (i.e., build competence). The professionalism tasks help you to make and communicate your plan. **Professionalism tasks have strict due dates that are not flexible.** *You cannot make up for a missed status professionalism task.* These tasks are relatively simple, and the main point is to complete them on time. Cutting these tasks will force you to earn points in other competence tasks, just like a professional make up for mistakes by working harder to prove their worth!

Attendance

You are not required to attend any individual sessions, but you should attend every lab/class session you can. D2L provides details on all of the class content, including recordings of each class (assuming no technical errors), but

attending synchronous sessions will give you a chance to get direct feedback from the teaching team and peers. Much of our time together will be dedicated to collaboration and feedback, thus hard to replicate later.

Unlimited Resubmissions

Failure is always an option but is never final so long as you go back and correct your mistakes. I hope all my students score perfectly on each assignment (it makes grading much easier!), but it is an unrealistic and unfair expectation. Challenging materials often require multiple attempts to master. Also, when we focus on 'correct' answers, we easily forget that correcting our mistakes is vastly more important than finding correct answers. Learning computing is about practicing programming processes, not achieving a specific solution. Your value is finding answers to never-before-solved problems, not memorizing answers someone already completed!

You are not just allowed ***but expected*** to correct your mistakes and resubmit your work. Identifying our mistakes is so important that I require a 'Cover Sheet' for most assignments. The Cover Sheet asks you to reflect on your learning process and how completing each assignment improves your overall skillset. See D2L for more details on Cover Sheets, but know that there will be automated checks to make sure you are submitting these with at least some consideration, and you may lose credit until you provide thoughtful consideration.

Interview As Assessment

Instead of traditional classroom testing, I use a few different ways to assess your growing skills. Tests have their purpose, but they seldom reflect the nature of computing tasks in the workplace. The main way that employers decide who to hire is through face-to-face interviews. To get high-quality jobs, you need to know how to do *and*

talk about computing. To that end, I use one-on-one interviews as a way of practicing this skill and ensuring you are sufficiently learning.

The interviews are only 15 minutes and will not be 'oral exams'. You will learn more about this process during the term, and the first interview is entirely practice. Preparing for these interviews is simple: complete the assigned work and reflect on 'good' answers when writing detailed Cover Sheets. If you put in the effort, report your progress honestly, and reflect on stories about the challenges and opportunities you meet in this class, the interviews will be easy.

Academic Honesty

Given my philosophy and approach, I hope there is little reason to 'cheat' since learning this invaluable material is the best way to succeed in the course and for your career. I believe that everyone wants to learn and can when given a chance. Sometimes, though, life tempts us towards shortcuts. I implore you not to do this. Taking shortcuts does not pay in the short or long term, and I believe I have several ways to detect and thus deter such activities.

The most common violation of academic honesty is taking the work of others. You have my permission to work alongside others in this class so long as you list your study partners on each submission's Cover Sheet and turn in your own original work. Most code submissions will also require you to submit a 'ledger' of your work. The test cases for the assignment will automatically capture your work and track your progress. The ledger documents the effort and original thought you put into the work or exposes last-minute attempts to avoid 'putting in the work'. Again, the best way to learn and succeed is to plan and put in the time. The ledger also provides an invaluable tool for getting help! It will help me see your thought processes and

perhaps help you get past any 'mental blocks' or other struggles you face. I am here to help you and would rather answer 'dumb' questions than have an uncomfortable conversation later about academic misconduct. I hope never to exercise my academic honesty clause. I will, if required, with the knowledge that I have given you every opportunity to learn without taking dishonest shortcuts.

Classroom Management and Policies

I Trust You

My policies seek to treat you as a professional. You choose how and when to complete coursework. I do not set strict deadlines, as I do not see myself in a position to set your priorities. If you have other things more important than this class, you are free to adjust times and levels of effort. You can work ahead or catch up. At times, I may check in on you, but I am not here to 'keep you on task'. If you are concerned with your time management, seek out the Personal Accountability Contract on D2L and follow its instructions. You can help yourself by working with peers (find your own or use your assigned groups in class). Above all, I trust you to make the best choices for your academic, professional, and personal life, but I am glad to help guide your options if it helps!

Getting Help

I expect you to reach out to any resources that help in your learning. You can ask peers, former students, family, friends, tutors, and most importantly, the instructional team for help. It is your responsibility to document what support you receive, but beyond that, you know what you can and cannot do independently. Try to shift your mindset away from "getting the right answer before the due date". Instead, think in terms of "how can I find someone to help me to work through this!"

While the traditional view of teachers is that of lecturers, the best help I can provide is teaching you to see where you are going wrong. I want to see your messy code! To be as accessible for this as possible, and given the challenges of continued social distancing, you can reach me two ways outside class for one-on-one aid.

In D2L, you will find a link to sign up for time slots. These will mostly be virtual, but some blocks will be available on the days we meet. This tool allows you to set a time for discussion.

You can also reach out to me at any time via Discord messages or email. I am often up late and glad to dive into a discussion or respond to a quick query. If you need help, start by sending me your current work (probably code) and your current issue as you see it. Often a quick reply will show you the way, but I am glad to find the time to work with you too for those pesky bugs!

Safety and Respect

One non-negotiable request is to respect the policies, statutes, laws, rules, or mandates that any recognized authority adds to the classroom or virtual environment. Any breaches of these will be reported to the school as appropriate and may lead to your removal from the class if not other University ramifications.