

CSPB 2270 - Summer 2023 Jones - Data Structures & Algorithms

[Dashboard](#) / [My courses](#) / [2234:CSPB 2270](#) / Welcome to CSPB 2270 - Data Structures & Algorithms

/ [Syllabus - CSPB 2270 Data Structures \(Summer 2023\)](#)

Syllabus - CSPB 2270 Data Structures (Summer 2023)

Instructor Information

- Name: Dr Frank Jones
- Email: francis.jones@colorado.edu
- Office hours:
 - Mondays 7:00 pm - 8:00 pm MT
 - Wednesdays 1:00 pm - 2:00 pm MT
 - or by appointment

Communication Channels

method	what types of communications use this method
Piazza	Should be used for all communications for the course. Piazza will be used to communicate all course information. You can send public messages to the entire class or you can send private messages to the instructors. (link will be available first day of classes)
Zoom	Will be used for Office Hours and Interview Grading meetings. (links will be available first day of classes)
Email	Please only use Email if you have an emergency and cannot access Piazza <i>OR</i> you need immediate assistance during exams. Make sure to include [CSPB 2270] in the subject of the message. For all other communications contact me through Piazza private messages.
Individual Check-in	Meetings via Zoom can be scheduled between student and instructor when you have a unique situation that may be affecting your work or class experience. You and the instructor can discuss the specific situation and develop a solution and/or strategy to move forward. (see Individual Check-in section below)

Important Dates

Assessment	Date
Exam 1	July 7th, 10am - 10pm MT
Exam 2	August 4th, 10am - 10pm MT
Interview Grade for Linked List Assignment	Jun 14-16
Interview Grade for Sorting Assignment	Jul 5-7
Interview Grade for Graph Assignment	Jul 26-28
Final Project	Aug 14-15
Submission due by Aug 14th.	
Interview Grading is on Aug 14th or 15th.	
Signups for a 20 minute interview slot will be available at the beginning of August)	
Quizzes	usually due on the Monday following the week the quiz is assigned
Programming Assignments	usually due on Tuesdays
Assignment Interviews	usually held on Wednesdays and Thursdays

Grade Breakdown

The final grade is distributed among reading quizzes, programming assignments, exams, and the final project, with the following distribution:

item	% of grade	notes
Reading Quizzes	10%	assignments from the textbook
Programming Assignments (10):	25%	Autograded
Assignment Interviews (3):	15%	All the Programming Assignments have an automated grading component (see above). For a few select assignments, there will also be an Interview grade. During the interview, you must be able to explain every detail of your submission as well as related questions to the topic. During the interview, your code must be able to receive the highest score recorded by the automated grader.
Exams(2)	30%	Includes multiple choice, open ended questions, and programming questions for you to show your command of the concepts.
Final Project	20%	The Final Project does not have an autograded portion. All the grading is via an interview grading session. Similar to the programming assignment interviews, you must know your code and be able to explain concepts while demonstrating the working code.

Extensions and Extra credit activities

There are no extensions granted on due dates for assignments and labs. There will be opportunities for extra credit activities which can compensate for some of your homework grades. Make sure to start working on the homeworks early in the week to avoid any last-minute issues. There are no extensions granted for quizzes.

Proctorio and exam requirements

Proctorio and exam requirements are posted in each course at the top of Moodle. Non-compliance with Proctorio, exam or course instructions may result in a voided exam (score = 0) or an honor code violation. Students are required to understand the conditions allowed for each exam. Please contact your instructor if you are unsure about requirements.

During exams you are allowed to use 1 page of scratch paper if you want but once the exam starts you must show your scratch paper (front and back) to the camera at least for 5 seconds each side. Make sure that your page fits in the camera frame by checking the image on your screen. Once you are done with your exam make sure to show your paper again for at least 5 seconds to the camera before you close your exam.

Grading

Letter grades will be assigned using the standard grading scale used in most courses in the Computer Science. Grading is based solely on an individual basis. The performance of your classmate will have no bearing on your own grades.

Grading Scale

score	grade	score	grade	score	grade	score	grade
		87%-90%	B+	77%-80%	C+	67%-70%	D+
93%-100%	A	83%-87%	B	73%-77%	C	63%-67%	D
90%-93%	A-	80%-83%	B-	70%-73%	C-	60%-63%	D-

Programming Requirements and Course Prerequisites

The prerequisite for this course is Introduction to Programming CSPB 1300 or similar. You are expected to understand the fundamentals of computer programming and have the abilities to seek out resources about specific languages, routines, and functions as needed.

In this class, our language of choice is C++ because it is the best (why?). Our students come from many different backgrounds and your intro to programming course may or may not have used C++. If you haven't worked with C or C++, a supplemental text is available. As a computer scientist you will often find you must learn new aspects of languages, and new tools as you go and there is never one set of rules that covers everything. There are many resources online about this language and almost always the first google link will give the answer you are looking for. If you have less experience with C++ then I highly recommend the ZYbook C++ optional text.

All students are different. Some find their experience in CSPB is enough to get them going more advanced topics in C++. Others find they need considerably more time to research and explore new tools. This course is not a coding bootcamp so be sure to set aside some time to sharpen your C++ skills in a manner best suited to your learning style.

Textbook and other Course Material

See week 0 for current textbook information.

Steps to purchase the class textbook are as follows:

- Click on a zyBooks link in Moodle for any week with textbook reading assignment (Do not go to the zyBooks website and create a new account)
- Follow instructions to Subscribe

Textbook includes quizzes that will be graded and included in your "quiz" part of your final grade.

Here are some helpful websites if you want learn more about C++ or to gain more experience in programming C++:

site	description
www.learncpp.com	This is a well-written, comprehensive online textbook aimed at true beginners.
cppreference.com	Encyclopedic reference, useful for looking up details but not learning high-level concepts.
cplusplus.com	Tutorials for those who already understand how to program (e.g. not for true beginners).
stackoverflow.com	This is a question & answer site that will have answers to any question a beginner might have (in varying levels of helpfulness). Please don't ask for course help there :)

You will need a Github account for accessing the programming assignments.

For this course and future courses, it is highly recommended that you use some method to save your work. Computer crashes happen frequently and you are responsible for saving and keeping back-up copies of your work. There are manual methods that might be used such as a Dropbox account, a Google Drive account, or simply a USB memory stick. However these method require a great amount of discipline by the programmer to be effective.

We recommend that you learn to use Git to store your work. When accessing a programming assignment, you will create a private repository that can be used to store intermediate versions of your files. The repository is accessible by the instructors which allows them to access your latest code versions when you are having programming issues. Git is the industry standard for source code management, and you will likely use it on a daily basis after you graduate (as well as for future courses).

There are only a few Git commands that you will need to use on a regular basis:

- CLONE creates a new copy of a given repository. You will use this when you accept the repository for a new assignment.
- STATUS provides a list of the current state of file changes for the local repository. It will list the files that have been modified and the files that you had tagged to be saved in the next COMMIT (see below).
- ADD the changed file to the list of files to save in the next COMMIT (see below).
- COMMIT the changes to the local repository. This command records all the changed files that have been tagged. The local repository is stored on the local machine (either your csel.io virtual machine or your local laptop/desktop) but has not been copied to the GitHub server.
- PUSH your local commits to the GitHub server. After this command completes successfully, your file versions are available to anyone that can access your repository. This mechanism allows the instructors to access your code.

Course Information

The course covers techniques for writing C++ code that implements various data structures and algorithms and to explore foundational topics in computer science in a practical, applied manner. This class assumes incoming students have completed CSPB 1300 or have equivalent programming experience, especially with C++.

Course Structure

Most weeks are about new content, some are for tests, and some are for working on the course project. Each week is designed to either relay some critical topic (e.g. smart pointers) for which you'll learn about via lectures and then have a homework assignment to implement a data structure or algorithm that explores that topic (e.g. for pointers, build a linked lists data structure).

There are two exams of equal weight and a final project. The project is your chance to make something interesting that shows off your masterful command of data structures and algorithms in C++.

Below you will find a list of the topics covered for each sequence.

Content Overview

This course will introduce you to the common methods of storing and accessing data. These method will be used throughout your career in different ways and at different levels of abstraction. In this course we assign programming assignments that implement the basic algorithms used throughout Computer Science. Gaining experience with the fundamental data structures will allow you to choose the correct methods in your future courses and career.

Sequence on Pointers

Understand the relationship between memory, values, and how memory addresses can be used as values.

Programming Assignment: Linked lists.

- Introduction to pointers
- Linked lists
- Stack and heap memory
- Double pointers
- Visualizing memory usage
- Code contracts (pre, post-condition, invariants)
- Linked list operations / homework help

Sequence on Recursion

Learn why and how functions can call themselves.

Homework: Binary search trees.

- Recursion overview
- Trees
- Binary search tree data structure
- Tree traversal methods
- Binary search tree operations
- BST remove
- Balanced trees
- Red black trees: insert operation

Sequence on Computational Complexity How can we measure how much of some resource (like time) does an algorithm take? Homework: sorting algorithms.

- Complexity & Big Oh notation
- Complexity of BST operations
- Hard Problems (e.g. traveling salesman)
- Bubble Sort
- Merge Sort
- Quick Sort

Sequence on Design What are some ways that we can approach designing and implementing very tricky things? Like a B-Tree data structure?

Homework: the B-Tree implementation.

- Software design
- Testing as a design strategy
- Operation times
- B-Tree overview
- B-Tree invariants
- B-Tree find operation
- B-tree demo
- B-tree insert operation
- B-tree remove operation

Sequence on Abstract & Concrete Data Types What are ADTs, what are some examples, and how do they relate to concrete implementations?

Homework: Priority queues.

- Abstract vs concrete data types
- Common methods (e.g. query, add, remove)
- Priority queues
- Treaps

Sequence on Hashes and Hash Functions How can we use mathematical hash functions to design and build more efficient data structures?

Homework: Hash table.

- Overview of hash functions
- Implementing hash functions
- Hash collision mitigation strategies
- Open addressing
- Hash tables, hash sets



Sequence on Encoding and Decoding Data How can we encode and decode data, and what are application areas for this?

Homework: Huffman coding

- Data coding
- Lossless text compression
- Huffman: introduction and how to build a Huffman tree
- Encode and decode using Huffman trees

Sequence on Graphs and Graph Algorithms Learn about applications for graph data structures, various types of graphs, and several algorithms for traversing them.

Homework: Depth- and breadth-first search.

- Introduction to graphs
- Implementing graphs
- Graph structure & metadata
- Social network example
- Depth-first search
- Breadth-first search
- Directed acyclic graphs
- Spanning trees

Course Outcomes

By the end of this course, students should have a solid grasp of the concept of data structures, algorithms that use them, and related concepts such as recursion, computational complexity, and how memory works. In particular, students will:

- Understand what data structures and algorithms are
 - Create C++ struct variables from scratch
 - Write functional-style algorithms and full programs that manipulate those structures
 - Be able to work with variables by value, by pointer, and by reference
 - Come to use recursion as second nature
 - Understand the utility of abstract data types, and how this differs from concrete data types
 - Be able to determine what an algorithm's computational complexity is
 - To learn a number of useful, common data structures; when, why and how to use & implement. These include lists, trees, and graphs.
 - Have an intuitive understanding of when a particular data structure or algorithm is appropriate, or not
 - Be able to debug, maintain, and improve data structures and algorithms
 - Have a foundation for proceeding to future courses such as Algorithms and Systems
 - Properly understand and frame problems before designing and implementing solutions
-
-
-

Applied Computer Science - Post-Baccalaureate - Syllabus Statements

Updated: December 2022

Individual Check-In

If you have a unique situation that may be affecting your work or class experience -or- you need clarification of an email, Piazza, or ZOOM exchange, please email your instructor with the subject line "Individual Check-In". Your instructor will email you back to set up a Zoom call to discuss the specific situation and work with you to develop a solution and/or strategy to move forward.

Service Interruptions and Support

Due to the online nature of the program, there is always the possibility of service interruptions. If you are unable to access the course materials (Moodle, Piazza, etc), we encourage you to visit <https://www.isitdownrightnow.com/applied.cs.colorado.edu.html>.

For non-urgent issues relating to platform support, please contact cscihelp@colorado.edu.

For urgent issues, for example, attempting to upload a homework assignment near a deadline, email your instructor directly. In cases of documented exceptional illness or circumstances affecting assignments, instructors may or may not offer an alternative assessment (which may differ in form or content) at the instructor's discretion. Please consult the <https://www.colorado.edu/registrar/students/withdraw> campus policy for medical leave, if needed.



Collaboration Policy

We welcome collaboration! Sharing insights, asking questions, learning by doing, and learning by helping others are essential skills in learning computer science.

Collaboration is discussing ideas of the course with others, sharing insights and extra resources, working through similar questions to an assignment, sharing resources, and helping others. The Piazza forum in your class is an ideal place to share ideas, lead a discussion or be the hero that asks the “dumb question” everyone else is afraid to ask. And Piazza is often a source for content for instructors to include in letters of recommendations. Your leadership, courage, and determination will not go unnoticed.

Collaboration is not:

- “Having a partner.” In particular, Group Projects, or projects that specify “working with a partner” will have individual guidelines.
- One student solving problems 1-4, and another solving 6-10.
- An identical group solution submitted by multiple students.

Unless specified in the assignment, all submitted coursework is individual. In general:

- You must document resources and collaboration on any assignment. This should be in the form of comments at the start of code and/or within solution notes.
- Cite Your Sources: If you collaborated with someone on an assignment, or if your submission includes quotes from a book, a paper, or a web site, you must clearly acknowledge the source.
- Plagiarism is forbidden. Copying answers directly or indirectly from solution manuals, web pages, or your peers is a violation of honor code. The assignments and code that you turn in should be written entirely on your own.
- Copying/soliciting a solution to a problem from the internet or another classmate constitutes a violation of the course's collaboration policy and the honor code and may have serious consequences.
- You may not actively search for a solution to the problem from the internet. This includes posting to sources like StackOverflow, Reddit, Chegg, CourseHero, etc.
- StackExchange Clarification: Searching for basic techniques in Python/C++ is totally fine.
- If you have taken this course prior to this semester and have done some/all of homeworks previous code or previous homework solutions may not be reused. You must start each homework from scratch.

When in doubt, ask. If something doesn't seem right - you are not sure if you can use a resource or if you are feeling pressure to share a specific solution - please reach out to your instructor.

Note: Other information on the Honor Code can be found at www.colorado.edu/policies/honor.html and <https://www.colorado.edu/sccr/honor-code>.

Code Interview Grading

This should be included in course syllabus (and Week 0 perhaps) if you are using Interview Grading. Please add any specific requirements for your course or assignments and use this as a general reference.

Coding Interview Grading FAQ

What is Code Interview Grading?

A live interview with an instructor or TA to assess student understanding of submitted code, and the student's ability to explain concepts in the code. Typically an important component of a coding project grade. To be successful in Computer Science we must be able to communicate a solution to both human and electronic computers.

What are the goals of a Code Interview?

- To prepare students for future code interviews in industry.
- Allow students to show their depth of knowledge and the opportunity to showcase their work.
- Allow student to demonstrate skills to communicate complex concepts and code develop ability to envision changes, predict outcomes,
- Assess important learning goals such as the ability to modify and predict outcomes, debug code, and demonstrate computer science skills beyond the ability to produce working code.
- Confirm the code submitted is the student's original work.

What can I expect during a code interview?

- Just as in an industry code review, the student should be able to explain and reflect on any aspect of the development process and any of the code under review.
- Typically, students will be given guidelines of basic concepts, tools or learning goals assessed ahead of time. These may be implicit in the assignment instructions.



- Some questions may be in response to student's responses to earlier questions, such as being asked to respond more completely or with more depth.
- Since questions given are based on student responses, not all questions are provided ahead of time, and students will be asked different questions.

How are Coding Interviews fair? My code works, so I should get full credit.

- Computer Science is about the communication of solutions to problems, not just getting your code to work.
- Interviews allow you to show your knowledge of the concepts behind the code by communicating how the code is implementing given concepts.
- Being able to visualize changes required to the code to handle alternative situations shows your detailed understanding of the code and how it works.
- Like all grading, code interviews are based on performance, not hidden knowledge or potential.
- Each student's depth of understanding is different and therefore each interview will be unique based on the student's individual responses. The interviews are based on exploring the student's skills or knowledge based on the learning goals for the material covered.
- Individual questions give students the best opportunity to clarify their responses and demonstrate their abilities.
- These interviews are designed to help you succeed. They are opportunities to show off your knowledge.

Piazza, ZOOM, and Email expectations:

What is Piazza?

Like a classroom, it is a place to ask questions, discuss learning strategies, explore related topics, support your classmates, and contribute to the class. Like a classroom blackboard, all students must read and follow the instructor posts for course and content information, and announcements.

How do we use Piazza?

It is used for participation grading - see your class syllabus for details. It is a place to practice professional collaboration strategies. Instructors may not reply to all inquiries and let other students answer. Many instructors will use Piazza as a reference in the future for a letter of recommendation.

What Piazza is not?

A 24 hour helpdesk or answer forum. A robot tutor. A personal blog. A place for others to completely debug your code.

Posts and questions about classwork should be content specific and reflect effort on student's behalf. Sample questions that are not content specific and hence not appropriate: Please tell me how to do #6 Explain #6 I'm lost on #6, Help!

Sample questions that are content specific: I applied the technique from the video to #6, but I get an answer that is too large, could my loop be incorrect? In the video lecture, I understand the algebra steps in #6, but why is $0! = 1$? Is anyone else getting different answers to #6? It seems to depend on which method I use. (notice the student is not posting the solution)

Any post that interferes with fellow student learning is not acceptable.

Any posts, regardless of intent, that could cause abuse, obstruction, disruption, or interference with student learning, including posts that are disrespectful, aggressive, distracting or inappropriate, will be saved and deleted at the discretion of the course instructor. Such comments will be reported Student Conduct & Conflict Resolution. Please note that there are many ways a post can interfere with student learning. If you have doubts about your message, please edit. For example, sarcasm doesn't translate well into text and could be a problem. Additionally, calling out specific students' performance or telling others how to do/work around a problem (ie: Hardcoding or providing links to answers)

What are ZOOM Office Hours?

It is much like a classroom. It is a live web session to ask questions, explore further, discussion strategies, explore related topics and support your classmates and contribute to the class. Office hours are optional.

How do we use ZOOM Office Hours?

Discuss class content directly with the instructor and other students. Practice professional collaboration strategies. Instructors may use break out groups or/and guide discussions, at their discretion. Instructors decide which topics will be discussed based on what will optimize learning. Please wear clothes and be aware of your environment.

What are ZOOM Office Hours not?

Answer forum. Tutoring session. Instructors cannot help you completely debug your code. Instructor may not reply to all inquiries and let other students speak as appropriate.

Questions should be specific and reflect effort on the student's behalf. Sample questions that are not content specific and hence not appropriate. Tell me how to do #6 Please explain #6 I'm lost on #6, Help!

Sample questions that are content specific. I applied the technique from the video to #6, but I get an answer that is too large, could my loop my incorrect? In the video lecture, I understand the algebra steps in #6, but why is $0! = 1$? Is anyone else getting different answers to #6? It seems to depend on which method I use. (notice the student is not sharing the solution)

Any behavior that interferes with student learning or university activities is not acceptable.

Any behavior, regardless of intent, that could cause abuse, interference, obstruction, disruption, or interference with student learning, including comments that are disrespectful, aggressive, distracting, off-topic or inappropriate, will not be tolerated and may result in being muted, at the instructor's discretion. Such behavior will be reported to Student Conduct and Conflict Resolution. For example, dominating group discussions can inhibit others opportunity to learn and will not be allowed.

What is CU Email?

Your CU email account is the official university method of communication. Enable settings in your account to ensure you see and read all email. Set aside a time each day to read email.

University of Colorado Boulder Syllabus Statements

updated: December 2022

Classroom Behavior

Both students and faculty are responsible for maintaining an appropriate learning environment in all instructional settings, whether in person, remote or online. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. For more information, see the [classroom behavior](#) policy, the [Student Code of Conduct](#), and the [Office of Institutional Equity and Compliance](#).

Requirements for COVID-19

As a matter of public health and safety, all members of the CU Boulder community and all visitors to campus must follow university, department and building requirements and all public health orders in place to reduce the risk of spreading infectious disease. CU Boulder currently requires COVID-19 vaccination and boosters for all faculty, staff and students. Students, faculty and staff must upload proof of vaccination and boosters or file for an exemption based on medical, ethical or moral grounds through the MyCUHealth portal.

The CU Boulder campus is currently mask-optional. However, if public health conditions change and masks are again required in classrooms, students who fail to adhere to masking requirements will be asked to leave class, and students who do not leave class when asked or who refuse to comply with these requirements will be referred to Student Conduct and Conflict Resolution. For more information, see the policy on classroom behavior and the Student Code of Conduct. If you require accommodation because a disability prevents you from fulfilling these safety measures, please follow the steps in the "Accommodation for Disabilities" statement on this syllabus.

If you feel ill and think you might have COVID-19, if you have tested positive for COVID-19, or if you are unvaccinated or partially vaccinated and have been in close contact with someone who has COVID-19, you should stay home and follow the further guidance of the Public Health Office (contacttracing@colorado.edu). If you are fully vaccinated and have been in close contact with someone who has COVID-19, you do not need to stay home; rather, you should self-monitor for symptoms and follow the further guidance of the Public Health Office (contacttracing@colorado.edu). Contact your instructor as soon as possible about your condition to arrange extension if needed.

Accommodation for Disabilities

If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to your faculty member in a timely manner so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities in the academic environment. Information on requesting accommodations is located on the [Disability Services website](#). Contact Disability Services at 303-492-8671 or dsinfo@colorado.edu for further assistance. If you have a temporary medical condition, see [Temporary Medical Conditions](#) on the Disability Services website.

Preferred Student Names and Pronouns



CU Boulder recognizes that students' legal information doesn't always align with how they identify. Students may update their preferred names and pronouns via the student portal; those preferred names and pronouns are listed on instructors' class rosters. In the absence of such updates, the name that appears on the class roster is the student's legal name.

Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the [Honor Code](#). Violations of the Honor Code may include, but are not limited to: plagiarism, cheating, fabrication, lying, bribery, threat, unauthorized access to academic materials, clicker fraud, submitting the same or similar work in more than one course without permission from all course instructors involved, and aiding academic dishonesty. All incidents of academic misconduct will be reported to Student Conduct & Conflict Resolution (honor@colorado.edu); 303-492-5550). Students found responsible for violating the [Honor Code](#) will be assigned resolution outcomes from the Student Conduct & Conflict Resolution as well as be subject to academic sanctions from the faculty member. Additional information regarding the Honor Code academic integrity policy can be found on the [Honor Code website](#).

Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

CU Boulder is committed to fostering an inclusive and welcoming learning, working, and living environment. University policy prohibits sexual misconduct (harassment, exploitation, and assault), intimate partner violence (dating or domestic violence), stalking, protected-class discrimination and harassment, and related retaliation by or against members of our community on- and off-campus. These behaviors harm individuals and our community. The Office of Institutional Equity and Compliance (OIEC) addresses these concerns, and individuals who believe they have been subjected to misconduct can contact OIEC at 303-492-2127 or email cureport@colorado.edu. Information about university policies, [reporting options](#), and support resources can be found on the [OIEC website](#).

Please know that faculty and graduate instructors have a responsibility to inform OIEC when they are made aware of any issues related to these policies regardless of when or where they occurred to ensure that individuals impacted receive information about their rights, support resources, and resolution options. To learn more about reporting and support options for a variety of concerns, visit [Don't Ignore It](#).

Religious Holidays

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, contact your instructor at least one week in advance to make arrangements to work around the conflicts.

See the [campus policy regarding religious observances](#) for full details.

Last modified: Wednesday, 17 May 2023, 9:11 AM

You are logged in as Taylor Larrechea (Log out)

[Data retention summary](#)

[Get the mobile app](#)

cscihelp@colorado.edu

