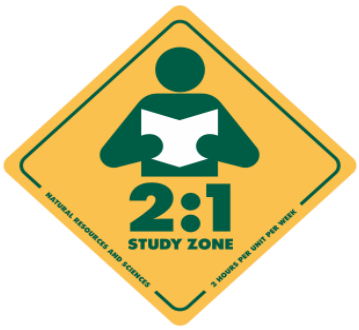
Cal Poly Humboldt
Course Syllabus for CS 211
Fall 2024

Data Structures



Basic Course Information

Instructor: **David Tuttle** **david.tuttle@humboldt.edu**

Lectures: **Tue & Thu 1:00 PM – 2:20 PM**
Natural Resources 101

Labs: **Mon 11:00 AM – 12:50 PM**
1:00 PM – 2:50 PM
BSS 315

Office Hours in BSS 326: **Mon/Tue/Thu/Fri 9:00 AM – 10:30 AM**

Online meetings via Zoom require prior arrangements

Course Web Page

Go to http://canvas.humboldt.edu to gain access to Humboldt's Canvas learning management system. Login using your Humboldt username and password. A link to the appropriate Canvas course page should be available from your home screen.

Course Description

Building upon the material from CS 112 - Computer Science Foundations II and MATH 253 – Discrete Mathematics, this course covers classic data structures and some basic algorithms used for solving problems involving structured collections of data. Data structures will be discussed from both the point of view of someone using them and the point of view of someone implementing them. Students will learn how to compare the performance and space requirements for different algorithms via the use of big-O notation, and learn how to evaluate the trade-offs between different approaches to determine which data structure(s) and algorithm(s) are appropriate for a given problem. Topics include discussion of primitive data structures such as arrays and linked lists, higher-level data structures such as stacks, queues, trees, graphs, and hash tables, and concepts such as big-O notation and recursion. This course lays the foundation for more in-depth discussions of algorithms, more advanced data structures, and algorithmic analysis presented in CS 312 - Algorithms.

Course Objectives and Learning Outcomes

This course addresses departmental learning outcomes of:

- Computational Thinking
- Technical Writing
- Communicating and Collaborating

This course addresses computational thinking at a moderate level, adding the next level of computational maturity through its coverage of classic data structures and algorithms. It addresses technical writing and communicating at a moderate level via program documentation and coding standards that stress reusable code. Furthermore, after successfully completing this course, students should be able to:

- Describe how the data structures in the topic list are allocated and used in memory, compare alternative implementations with respect to performance, and describe common applications for each.
- Write programs that use each of the following data structures: arrays, linked lists, stacks, queues, trees, graphs, and hash tables.
- Choose the appropriate data structure for modeling a given problem.
- Describe the concept of recursion and give examples of its use, identify the base case and the general case of a recursively-defined problem, and implement, test, and debug simple recursive functions and procedures.
- Explain the use of recurrence notation to describe the amount of work done by an algorithm; determine the time and space complexity of simple algorithms, and implement the most common $O(N^2)$ and O(N log N) sorting algorithms.

Humboldt Learning Outcomes that this course addresses:

This course explicitly contributes to students' acquisition of skills and knowledge relevant to Humboldt Learning Outcomes. Humboldt graduates will have demonstrated:

- Effective communication through written and oral modes.
- Critical and creative thinking skills in acquiring a broad base of knowledge and applying it to complex issues.
- Competence in a major area of study.
- A preparedness to succeed in their chosen careers.

Course Prerequisites/Co-requisites

Students enrolled in CS 211 are required to have successfully completed: CS 112 or equivalent; and MATH 253 or equivalent. Alternate prerequisites and co-requisites may be permitted with the explicit permission of the instructor and CS academic adviser.

Full knowledge of and familiarity with the C++ language to the extent covered in CS 112 is required!

Required Course Text, Materials, etc.

- **REQUIRED: Data Structures with ZyLabs,** an online text published by Zybooks. Available via the Canvas page in the Assignments and Modules sections. It's important to access this text through the appropriate Canvas assignment/module so that you automatically receive credit for assignment work done in the online text! **NOTE: The ZyLabs are <u>new</u> in CS 211 for Fall 2024.**
- **RECOMMENDED: Problem Solving with C++**, by Walter Savitch. The latest edition is the 10th, but any edition from the 5th on will suffice for this course. This book is a good reference for the C++ language.
- **REQUIRED: An echo360 PointSolutions (formerly Turning Technologies) license**

  If you already have a license from previous courses that is still valid through at least the end of the Fall 2024 semester, you need only register your account by clicking on the "Turning Account Registration" link on the Canvas course home page and following the instructions.

  If you do not have such a license, you can purchase one by clicking the "Turning Account Registration" link on the Canvas course home page and following the instructions. We recommend buying a longer-term license, as this software is used in several courses throughout the Humboldt CS curriculum. One license covers clickers in all CS classes for the period you purchase.

  This license will allow you to answer Clicker Questions asked during lectures starting in Week 2. Clicker Questions will be asked <u>only</u> during lectures.
- **RECOMMENDED: The PointSolutions app on an iOS or Android device (or you can use Web-based access through the echo360 website). The app itself is free to download, after you purchase the license.**

  A physical clicker is not required. If any student still needs to use a physical clicker, this can be accommodated ONLY by request, but it is expected that students will use the app or the Web interface to answer clicker questions.
- **REQUIRED: The free CS50 IDE (Interactive Development Environment) which supports C++ labs and assignments during the semester**

  CS50 was developed by Harvard for students to compile and run source code without having to install software on their workstations. CS50 has ties to the GitHub source code management system and has an interface based on Microsoft's Visual Studio. CS50 is free to students who sign up using their **humboldt.edu** email address, which we highly recommend.

  It is expected that students who have reached CS 211 are familiar with at least one IDE to do their programming. Students are free to use a different IDE when working on their homework and lab assignments, BUT the code should then be uploaded to and tested on the CS50 environment before submission, as that environment will be used to grade assignments and lab work. <u>ONLY</u> CS50 will be used to grade assignments!

  Students may run into some compatibility issues with the CS50 environment, especially if they have become accustomed to features and syntax specific to other IDEs. CS50 will require writing code that does not make use of such IDE-specific features, so be sure to leave sufficient time when doing assignments to ensure you do not overly rely on such features.

  The CS50 environment is available at this link: **<u>cs50.dev</u>**

Clickers

This class will ask clicker questions and expect student responses starting in the 2nd week of the semester! There will be multiple-choice "clicker questions" during class. Each question will be asked *twice* – once to get individual responses, then again after a short discussion period among student groups. ONLY the 2nd response will be used for your clicker grade! You will receive **2 points** for each correct answer, **1 point** for each incorrect answer, and **0 points** for no answer.

Full credit for the clicker-question portion of the semester grade will be set to **80% of the maximum possible**. This allows for the possibility of missing several clicker questions (or even a few class sessions) without penalty. For example, if 90 points' worth of clicker questions are asked over the course of the semester, a total of 72 points will be enough to receive full Clicker Question credit for the semester. "Full credit" is the ceiling – extra credit is not available if you score more than 80%.

Clicker questions will help you to see if you are starting to understand the concepts under discussion; sometimes they will provide review of concepts as well. Clicker questions allow you to get some immediate feedback about course concepts, and whether you need to pay more attention to course discussions and readings. **Using another student's clicker, or having someone else use your clicker, is considered cheating, with the same consequences as turning in someone else's work as your own or permitting someone to copy your work.**

Grading

Your semester grade will be determined based on the following percentages:

Online Text Activities and online ZyLabs:		20%	
Instructor-Assigned Programming Projects:		20%	
Weekly Lab Sessions:		20%	
Clicker questions:		10%	Full credit for getting 80% of all possible clicker points
Exams:	**Exam #1:**	10%	Given after 5 weeks of the semester
	Exam #2:	10%	Given after 11 weeks of the semester
	Final:	10%	**TUESDAY 12/17/2024 @ 12:40 PM**

In addition, the instructor may, at his discretion, issue a **non-passing semester grade** for the course if participation in **any** aspect of the class falls below a minimum of 50% of the total possible points for that part of the class. For example, if missed homework assignments result in an average homework score of less than 50%, then a semester grade of D or F may be assigned, even if the overall grade average is above 70%. The purpose of this rule is to prohibit the practice of simply ignoring part of the course requirements with the thinking that the other parts will be enough to pass. Participation in all aspects of the course must be maintained at acceptable levels in order to learn this material!

Your semester letter grade will be determined based on this chart:

Overall Percentage	Letter Grade	Overall Percentage	Letter Grade	Overall Percentage	Letter Grade
		93.0 and up	A	90.0 to 93.0	A –
87.0 to 90.0	B +	83.0 to 87.0	B	80.0 to 83.0	B –
77.0 to 80.0	C +	73.0 to 77.0	C	70.0 to 73.0	C –
65.0 to 70.0	D+	60.0 to 65.0	D	Less than 60.0	F

Expectations of the Student

1. **Attend Class Lectures, and Participate in Labs!**
2. Check your Humboldt email and Canvas Announcements **daily**.
3. Check the Canvas course website **frequently** for assignments, announcements and updates.
4. Expect to spend about 12 hours per week **outside of class** working on this course.
5. Learning through collaboration (defined as working with or learning from another) is an effective tool used in this class and in your future employment. When you are expected to collaborate, such as in weekly labs, it'll be made very clear in the assignment instructions. **All other work in this class is to be done independently. You may work together in study groups, but NO SHARING OF CODE IS EVER ALLOWED! Both the giver and the recipient of code are liable for penalties! (See the Plagiarism section below for more information)**
6. **Use of automatic code-generating programs is STRICTLY PROHIBITED!**

Class Culture

We will decide on the final expectations together, but some of the guiding principles might involve:

- Respect for each other (what does that mean to you?)
- Come to class and lab sober
- Keep cell phones and other distractions put away
- Be in the meeting before lab starts, so that you're ready when it starts
- Do not leave in the middle of lab unless there is a real need (e.g., family emergency, you are too ill to stay in class)
- Stay until your lab team has completed the lab work or until the end of the lab period, whichever comes first
- Be a regular and willing participant

Additional Coursework-Related Policies

Inclusivity

Students in this class are encouraged to speak up and participate in-class and online. Each of us must show respect for each other because our class represents a diversity of beliefs, backgrounds, and experiences. We believe that this is what will enrich all of our experiences together. We recognize that our individual differences can deepen our understanding of one another and the world around us, rather than divide us. In this class, people of all ethnicities, genders and gender identities, religions,

ages, sexual orientations, disabilities, socioeconomic backgrounds, regions, and nationalities are strongly encouraged to share their rich array of perspectives and experiences.

If you feel your differences may in some way isolate you from our classroom community or if you have a specific need, please speak with me early in the semester so that we can work together to help you become an active and engaged member of our class and community.

Plagiarism and AI Generative Tools

Plagiarism is a serious offense. Work submitted in programming assignments must be your individual work – no copying of others' work! Evidence of copying or plagiarism will result in appropriate penalties, up to and including a failing grade in the course. Students are responsible for knowing policy regarding academic honesty. Among the actions that are unacceptable are submitting another's program, code, or file as your own; giving programs, code, or files to another; and failing to quote material (that includes algorithms, project, code, and comments, too!) taken from another source. This even applies to comments as well as actual executing code!

Generative artificial intelligence (AI) programs, such as ChatGPT, may **not** be used for any work or assignments required in this course. The use of generative AI programs defeats the writing requirements and critical thinking skills that are vital to achieving our learning outcomes. Submission of partial or complete work from generative AI programs is not permitted and will be treated as plagiarism as defined in Humboldt's Student Academic Honesty Procedure, as outlined at policy.humboldt.edu/academic-integrity-and-honesty-policy .

Note that in the case of pair programming during weekly lab sessions, you are intended to work together and submit a single set of code, where one student types the code while all team members collaborate. **This is the ONLY authorized exception to the above policy!**

Late Submissions of Assignments

Programming Projects will be subject to a 10% deduction in credit if submitted after the due date and less than one week late. After one week, the deduction increases to 20%, and stays at 20% until solutions are posted or the end of classes, whichever comes first. After solutions are posted for a particular homework, submissions cannot be accepted for credit! Solutions may be posted just before the next exam, or, in some cases, no solution may be posted at all.
Online Text Assignments will have a posted due date and a late due date (usually one week later). After the late due date, the Zybooks software **will not issue any credit** for work done in the online text! Work done before the late due date will be fully credited, with no late penalty.
Lab Assignments should be completed by the end of the lab period, unless the instructor makes an exception for a particular lab. Lab attendance is **required** for lab assignment credit!
Clicker Questions will be asked during lecture, with no opportunity for make-up unless specific previous arrangements are made with the instructor, and only in case of an "official" scheduling conflict, such as university athletic events.

Students with Disabilities

Persons who wish to request disability-related accommodations should contact the **Student Disability Resource Center** in the Learning Commons of the Lower Library, **826-4678 (voice)** or **826-5392 (TDD)**. You can reach the SDRC's website at disability.humboldt.edu . Please note that some accommodations testing may take weeks to arrange, so students should contact the SDRC as soon as possible. Please do not request accommodations directly from me – the SDRC handles requests and contacts me on behalf of the student. **All SDRC-approved accommodation requests will be honored.**

Tutoring and Other Options for Assistance

The Learning Center in the Library may have resources available for tutoring in the topic of Data Structures. Contact the Learning Center for information. The instructor will also attempt to gather information about tutoring and other assistance options and announce them during lecture. The Learning Center's website is at learning.humboldt.edu .

Time Expectations

Remember the general rule of thumb for programming courses in Computer Science: *To be successful in a course in programming in Computer Science, you should plan to spend at least 3 hours outside of class for each 1 hour of college course credit. That implies an estimate of* **at least 12 hours a week spent outside of class for this 4-credit course**. You should be aware that:

- You can only learn programming by practicing it. Practicing programming as much as possible helps!
- This can include playing around with in-class examples, experimenting to see if something you are curious about really works like you think, and so on.
- Think of a musical instrument – you have to practice to master playing a guitar, violin, trumpet, drums, etc. You can't master it by just reading about the instrument. Think, also, of sports skills such as pitching, putting, etc. – again, repetition and practice is required to hone such skills.
- Writing programs can be a notorious time eater. Occasionally, a problem with code can potentially take large amounts of time to locate and fix (especially if you don't ask for help!).
- Starting early enough so that you have time to ask questions when you run into problems can help with this!
- Why spend 4 hours struggling with a frustrating roadblock the night before the assignment is due, when you can spend 10 minutes composing an e-mail early in the week, work on other problems while waiting for the answer, and then get a reply that makes everything clearer as soon as you read it?
- The course will intensify as the semester progresses – as you are able to do more, you will be expected to do more. Also, later concepts are built upon earlier concepts as the course progresses. If you ask me as soon as you realize that some concept is not clear to you, that can help keep you from falling behind.
- If you have not completed an assignment by the deadline, your best choice is to submit whatever you have managed to do by then, as partial credit is your friend.

Campus Policies and Procedures

In 2016, the Faculty Senate revised the course syllabus guidelines to allow instructors to simply provide a web link to Policies and Procedures and make that link a part of the syllabus. All the policies on this web page are applicable to this class, and you are expected to be familiar with these policies! (If this syllabus and the posted Syllabus Addendum say contradictory things about a course policy, this syllabus will be the correct interpretation of the policy for this course.) The link is: academicprograms.humboldt.edu/content/syllabus-addendum