CSCI-2270: Computer Science 2: Data Structures. Summer 2017

INSTRUCTORS

Austin Holler: Austin.Holler@colorado.edu, office hours T-W-Th, 11:40-12:40, ECCS 122
Shirly Montero: Shirly.MonteroQuesada@colorado.edu, office hours T-W, 1:00-2:30, ECCS 122

TEACHING ASSISTANTS, COURSE ASSISTANTS, AND GRADERS

TA: Abhijit Suresh: Abhijit.Suresh@colorado.edu
CA: Haris Khan: haris.khan-1@colorado.edu
CA: Justin Cai: justin.cai@colorado.edu
Grader: Chaitra Ramachandra: Chaitra.Ramachandra@colorado.edu
Grader: Ryo Suzuki: Ryo.Suzuki@colorado.edu

The complete office hours schedule for TAs and CAs is available on Moodle. Click on the link for “CSCI2270SU2017 Office Hours” at the top of the Moodle page.

BRIEF DESCRIPTION OF THE CONTENT OF THE COURSE

Studies data abstractions (e.g. lists, stacks, queues, trees) and their representation techniques (e.g., linking, arrays). Introduces concepts used in algorithm design and analysis including criteria for selecting data structures to fit their applications.

Requisites: Requires prerequisite courses of CSCI 1300 or CSCI 1310 or CSCI 1320 or ECEN 1030 or ECEN 1310 and APPM 1345 or APPM 1350 or MATH 1300 or MATH 1310 (all minimum grade C-).

LECTURES

Holler: M-Th 1-2:20 pm in FLMG 156.
Montero: M-Th 11-12:20 pm in FLMG 156.

TEXT AND RESOURCES

1. Hoenigman, R. 2015. Visualizing Data Structures. Lulu Press.
2. Various online C++ resources

OTHER MATERIALS

It is highly recommended that you get a Dropbox account, or invest in a USB memory stick, to save files created in the virtual machine environment. This environment is not backed up, and if it crashes, you will lose all of your work.

Course materials, such as lecture notes and assignments, will be available in electronic form on the Moodle site for the course: http://moodle.cs.colorado.edu/. The enrollment key is k4csci2270su17.

COURSE OUTCOMES

In this course, the student is introduced to different methods of structuring data, as well as, the trade-offs of different data structures in terms of memory requirements and computational complexity, when building and searching these structures. The course will be taught in C++. Upon completion of the course, the student can:

1. Document code including precondition/postcondition contracts for functions and invariants for classes.

2. Create and recognize appropriate test data for simple problems, including testing boundary conditions and creating/running test cases, and writing simple interactive test programs to test any newly implemented class.

3. Design and test new classes using the principle of information hiding for the following data structures: array-based collections (including dynamic arrays), list-based collections (singly-linked lists, doubly-linked lists, circular-linked lists), stacks, queues, binary search trees, hash tables, graphs, and at least one balanced search tree.

4. Identify features and applications of common data structures, including records/structs, lists, stacks, queues, trees, graphs, and maps.

5. Implement algorithms for standard operations on common data structures and discuss the complexity of the operations.

6. Comment on the features of different traversal methods for trees and graphs, including pre-, post-, and in-order traversal of trees.

7. Describe the implementation of hash tables, including algorithms for collision avoidance and resolution.

8. Describe the principles of recursion and iteration and implement recursive and iterative solutions for a problem.

9. Formulate and implement solutions to problems using fundamental graph algorithms, including depth-first and breadth-first search and a shortest-path algorithm.

10. Explain the features of at least one tree balancing algorithm and how tree balancing affects the efficiency of various binary search tree operations.

11. Evaluate the complexity of different algorithmic strategies, including divide-and-conquer and greedy, used in sorting algorithms (insertion, mergesort, quicksort)

12. Discuss the choice of different data structure for modeling a given problem.

13. Correctly use and manipulate pointer variables to change variables and build dynamic data structures.

14. Explain the differences between dynamic and static data structure implementations, and justify the use of static and dynamic implementations in different applications.

15. Explain design choices and algorithm features in small-group settings.

Here is a brief list of topics covered:

Introduction to C++, pointers, memory, classes and structs.
Arrays and Vectors
Linked lists, single and double
Stacks and queues
Trees
Graphs
Hash Tables
Searching, sorting and algorithmic complexity

GRADING

Recitation	30% (weekly, due in-class and by 5pm on Sunday)
Homework Assignments	40% (weekly, due by 8am on Friday and on Wednesday)
Project	10% (one, due the last week of class)
Quizzes*	20% (weekly in class, 7 in total)

*MUST receive a cumulative score of at least a 65 average on the quizzes OR YOU CANNOT RECEIVE BETTER THAN D+ in the course.

The grades for this class follow the standard percentage breakdown for the College of Engineering:
93%-100% A
90%-93% A-
87%-90% B+
83%-87% B
80%-83% B-
77%-80% C+
73%-77% C
70%-73% C-
67%-70% D+
63%-67% D
60%-63% D-
0%-60% F

ADDITIONAL QUIZ GRADE REQUIREMENTS TO PASS THE CLASS

There are seven quizzes in this class, administered on Wednesdays, which account for 20% of your final grade. There are 70 points possible on these exams, which break down to 10 points per quiz. You must earn at least 45.5 points on the seven quizzes to receive better than a D+ in the class, regardless of your scores on the other aspects of the class. The 45.5 points is equivalent to a 65% average on your quizzes. The quiz schedule is given in the class schedule shown below. A C- course grade is needed to take the next class in the computer science sequence.

The final exam could be used to replace your lowest quiz grade. If you attempt this exam, your score will replace your previous lowest score (even if the final exam is lower). If you are satisfied with your grades, you do not need to take the final exam.

RECITATIONS

Bringing your own laptop to recitation will be necessary, as the size of the course does not allow for using a computer lab. Recitations will consist of two parts: an in-class and a take-home portion. All in-class exercises are able to be completed during recitation and it will be expected to be so. The take-home work will be due Sunday 5pm, following the recitation. No late assignments for recitations will be accepted.

ASSIGNMENT GRADING LOGISTICS

Assignments in the class will be graded using one or a combination of the following: interview with your recitation TA, grading script or auto-grader. If you are not happy with the grade you receive from the auto-grader, you have the option of scheduling a grading interview with your recitation TA. Also, you may be randomly selected for each assignment to participate in a grading interview. In the interview, you will be asked questions about the assignment you submitted. These questions are designed to test your understanding of the code as well as provide you with an opportunity to ask questions of your TA. Please note that there is no guarantee that the grade you get in the interview will be a higher grade. Your TA will announce when grading meetings are available, and it is your responsibility to schedule an interview with your TA as soon as the scheduler is posted.

INTERVIEW GRADING GUIDELINES:

It is important that every student attends their grading meetings at the scheduled time. Please take the time to read and understand the following interview grading guidelines.

- Grading meetings are 10- to 15-minute appointments, scheduled Monday-Friday
- Sign-up is on Moodle
- The grade you receive for an assignment is 40% for the technical components and 60% for the interview. If you do not attend an interview, you lose the 60%.
- If you need to reschedule your interview, that is okay, but you must email your TA at least 24 hours in advance (i.e. no waking up 5 minutes before the appointment, realizing you are going to be late, and sending a panicked email cancelling at the last minute). Emergency situations are an exception and will be evaluated on a case-by-case basis.
- There is a 1-minute "grace period" for being late, after that it is 10% off for each minute the

student is late, at 6 minutes late you get a zero for the interview.
• Our advice to all students is to get to the appointment 5 minutes *early* and use the extra time to prepare.

HARD CODING POLICY

If you hard code, you will get a zero. There are no exceptions, as hard coding an assignment is not truly completing an assignment.

SUBMITTING WORK LATE

You can receive a one-day extension on any assignment or recitation for a 20% grade penalty. After one day, your homework is late and cannot be turned in. In the event of a documented personal, family, or medical emergency, consult your TA about receiving a penalty free extension. If you know you will be missing a recitation, you might want to go to a recitation with the same TA being held at a different time.

ATTENDANCE

Attendance at all class meetings and recitations is required to receive credit for any activities that occurred during the meeting.

In recitation, some activities will be due during recitation and they must be submitted during recitation to receive credit. You are responsible for knowing the material presented during class and recitation, even if you were not in attendance when the material was presented.

OTHER ASSIGNMENT INFORMATION

Written work must be neat and readable, with adequate spacing and margins. Your name, the date, and your section number must be at the top right of the first page. Code files should have your name, date, and homework number included as comments at the top of the file.

APPROXIMATE COURSE SCHEDULE (A LIVING THING…)

Date	Topic	Reading	Assignment	Recitation	Quizzes/Due
Week 1 - June 5-11	Introduction to Data Structures. Algorithms, Complexity, and Pseudocode. Arrays and Sorting Algorithms	Introduction, Chapter 1,3 (until 3.3), 4	Assignment 1 - C++ review, sorting	Recitation 1- VM install, exercise, and insertion sort	Recitation 1
Week 2 - June 12-18	C++ arrays and structs Dynamic memory allocation Memory and pointers Classes Introduction to Linked Lists	Chapter 2, 3 Linked lists: 5, 5.1, 5.2	Assignment 2 - Array doubling, structs, file I/O	Recitation 2 - dynamic memory, array doubling, pointers and linked lists.	Quiz 1 Assignment 1 Recitation 2
Week 3 - Jun 19-25	More on Linked Lists	Chapter 5 finish	Assignment 3- Linked lists	Recitation 3- Linked lists	Quiz 2 Assignment 2 Recitation 3
Week 4 - June 26 - July 2	Stacks, queues	Chapter 6, 7	Assignment 4- Queues	Recitation 4- stacks and queues	Quiz 3 Assignment 3 Recitation 4
Week 5- July 3-9	Binary Trees and BST	Chapter 8,9	Assignment 5 - BST	Recitation 5 - BST	Quiz 4 Assignment 4 Recitation 5
Week 6- July 10-16	Recursion Tree Traversal, Red-black Trees	Chapter 10, 11	Assignment 6 - RB Trees	Recitation 6 - Tree Operations	Quiz 5 Assignment 5 Recitation 6
Week 7 - July 17 - 23	Graphs, Breadth-first search, depth- first search, Dijsktra	Chapter 12	Project-Graphs	Recitation 7- Dijkstra	Quiz 6 Assignment 6 Recitation 7
Week 8 - July 24- 28	Hash Tables/Functions, Review, The C++ STL, Vectors	Chapter 13			Quiz 7 Project Final

CLASSROOM BEHAVIOR

Students and faculty each have responsibility for maintaining an appropriate learning environment. 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 differences of race, color, culture, religion, creed, politics, veteran's status, sexual orientation, gender, gender identity and gender expression, age, ability, and nationality. Class rosters are provided to the instructor with the student's legal name. We will gladly honor your request to address you by an alternate name or gender pronoun. Please

advise us of this preference early in the semester so that we may make appropriate changes to our records. For more information, see the policies on class behavior and the student code.
It is our expectation that each of you will be respectful to your fellow classmates and instructors at all times. In order to create a professional atmosphere within the classroom, you are expected to:

*Arrive to class on time.
*Turn off your cell phone (talk and text).
*Bring your laptop to class, if you have one, to participate in classroom activities. Please restrict laptop use to these activities only, no email, Facebook, Youtube, etc.
*Put away newspapers and magazines.
*Refrain from having disruptive conversations during class.
*Remain for the whole class; if you must leave early, do so without disrupting others.
*Display professional courtesy and respect in all interactions related to this class.

Compliance with these expectations will assist all of us in creating a learning community and a high quality educational experience.
Though many of the above stated policies address academic climate within the classroom, these policies should also be upheld outside of the classroom. As a member of the CU community you are expected to consistently demonstrate integrity and honor through your everyday actions. Faculty, TAs, and staff members are very willing to assist with your academic and personal needs. However, multiple professional obligations make it necessary for us to schedule our availability. Suggestions specific to interactions with faculty and staff include:

*Respect posted office hours. Plan your weekly schedule to align with scheduled office hours.
*Avoid disrupting ongoing meetings within faculty and staff offices. Please wait until the meeting concludes before seeking assistance.

EMAIL POLICY

Respect faculty and staff policies regarding use of email and note that staff and faculty are not expected to respond to email outside of business hours. Send email messages to faculty and staff using a professional format.
This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, we encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at: https://piazza.com/colorado/summer2017/csci2270/home

If you feel that you need to send us an email within piazza or directly to our email address, you must follow the following tips for a professional email or they will be ignored:

*Always fill in the subject line with a topic that indicates the reason for your email to your reader. (Use **CSCI2270** in your emails for this course)
*Respectfully address the individual to whom you are sending the email (e.g., Dear Professor Smith).
*Avoid email or text message abbreviations.
*Be brief and polite.
*Add a signature block with appropriate contact information.
*Reply to email messages with the previously sent message. This will allow your reader to quickly recall the questions and previous conversation.

CSCI 2270 SUMMER 2017 COLLABORATION POLICY

The Computer Science Department at the University of Colorado at Boulder encourages collaboration among students. To support students in collaboration the Department has created a Collaboration Policy that makes explicit when their collaborative behavior is within the bounds of collaboration and when it is actually academic dishonesty, and therefore a violation of the University of Colorado at Boulder's Honor Code.

Students are most successful when they are working with other students to understand new concepts. The ultimate goal is that you fully understand the code you develop and be able to collaborate with others in a mutually beneficial way.

Unless otherwise specified, you may make use of outside resources (internet, other books, people), but then you must give credit by citing your sources in the comments inside your code. Use of outside resources does not include downloading complete, or almost complete, solutions to an assignment, whether you cite the source of the solution or not. This is considered plagiarism and violates the University's Honor Code policy.

Examples of citing sources include:
// Modified version from https://github.com/Phhere/MOSS-PHP
// Adapted from Program #7.2 in book "Accelerated C++" by Stroustrup
// Worked with Joe Smith from class to come up with algorithm for sorting // Received suggestions from stackExchange website (see http://....)

A good rule of thumb: "If it did not come from your brain, then you need to attribute where you got it."

Collaboration Exceptions

Certain homework, quizzes, or exams may be required to be completed without outside resources (see course overview for details). In these cases it is your responsibility to know the extent of approved resources and use only those that have been specifically allowed. Use of outside resources in these cases would violate the collaboration policy.

Examples of violating the Collaboration Policy

- Sharing a file with someone else.
- Submitting a file that someone else shared with you.
- Stealing a copy of someone else's work and submitting as your own (even with modification).

- Copying or using outside resources to solve a component of a larger problem and not citing your sources.
- Copying or using an entire solution that you didn't generate, regardless of whether you cited your sources.

Examples of collaborating correctly

- Asking another student for a helpful suggestion.
- Reviewing another student's code for issues/bugs/errors.
- Working together on the whiteboard (or paper) to figure out how to approach and solve the problem. In this case you must include that person's name in your collaboration list at the top of your submission.

One way to know you are collaborating well is if everyone fully understands the code that is developed. If you do not understand what is in your code or why certain parts of the code are included, you need to ask someone to clarify! This collaboration policy requires that you be able to create the code (or solve the problem) on your own before you submit your assignment.

Any discovered incidents of violation of this collaboration policy will be treated as violations of the University's Academic Integrity Policy and will lead to an automatic academic sanction in the course and a report to both the College of Engineering and Applied Science and the Honor Code Council. Students who are found to be in violation of the Academic Integrity Policy can be subject to non-academic sanctions as well, including but not limited to university probation, suspension, or expulsion.

Other information on the Honor Code can be found at www.colorado.edu/policies/honor.html and www.colorado.edu/academics/honorcode. Collaboration boundaries are hard to define crisply, and may differ from class to class. If you are in any doubt about where they lie for a particular course, it is your responsibility to ask the course instructor.

ACCOMMODATION STATEMENT

We are committed to providing everyone the support and services needed to participate in this course. If you qualify for accommodations because of a disability, please submit to your professor a letter from Disability Services in a timely manner (for exam accommodations provide your letter **at least one week prior** to the exam) so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities. Contact Disability Services at 303-492-8671 or by e-mail at dsinfo@colorado.edu. If you have a temporary medical condition or injury, see Temporary Medical Conditions: Injuries, Surgeries, and Illnesses guidelines under Quick Links at Disability Services website and discuss your needs with me.

RELIGIOUS OBSERVANCES

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 assignments/attendance. If this applies to you, please speak with me directly as soon as possible at the beginning of the term.

SEXUAL MISCONDUCT, DISCRIMINATION, HARASSMENT AND RELATED RETALIATION

The University of Colorado Boulder (CU Boulder) is committed to maintaining a positive learning, working, and living environment. CU Boulder will not tolerate acts of sexual misconduct, discrimination, harassment or related retaliation against or by any employee or student. CU's Sexual Misconduct Policy prohibits sexual assault, sexual exploitation, sexual harassment, intimate partner abuse (dating or domestic violence), stalking or related retaliation. CU Boulder's Discrimination and Harassment Policy prohibits discrimination, harassment or related retaliation based on race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. Individuals who believe they have been subject to misconduct under either policy should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127. Information about the OIEC, the above referenced policies, and the campus resources available to assist individuals regarding sexual misconduct, discrimination, harassment or related retaliation can be found at the OIEC website.

HONOR CODE

All students of the University of Colorado at Boulder are responsible for knowing and adhering to the academic integrity policy of this institution. Violations of this policy may include: cheating, plagiarism, aid of academic dishonesty, fabrication, lying, bribery, and threatening behavior. All incidents of academic misconduct shall be reported to the Honor Code Council (honor@colorado.edu; 303-735-2273). Students who are found to be in violation of the academic integrity policy will be subject to both academic sanctions from the faculty member and non-academic sanctions (including but not limited to university probation, suspension, or expulsion). The Honor Code Office has more information.