

CS 2420 Algorithms and Data Structures

Instructor:

Dr. Vicki Allan, Vicki.Allan@usu.edu. Please send messages directly to me through email rather than using Canvas messaging

Office/Phone: 429 Old Main, 797-2022

Office Hours:

MWF: 10:30-12:30. Other hours by appointment. (Send Email to set up.)

Credits: 3, meeting MWF in Huntsman Hall 222. There are two sections: 1:30 and 2:30. This course is required for all majors.

Prerequisite: CS 1410 or equivalent knowledge. CS 2420 is a fun and challenging class. You should plan to spend ten hours per week on this class. I expect everyone to grow markedly because of this class.

If you feel that you are a poor C++ programmer, your chances of succeeding in this class are slim. I would strongly recommend that you (re)take CS1410 instead of this class. There is so much new material to be learned, you do not have time to learn material that you should have mastered in earlier classes. Sometimes students like to gamble on success. The laws of gambling state that the smaller the probability of succeeding, the greater the reward should be. Many times the gamble in taking a class (given the time you are willing to spend and the skills you bring to it) has a very low probability of success and a small reward (you may barely pass, but the hours required may cause you to do poorly in other classes). Every semester I see many students throwing away time and tuition money by taking classes they can't possibly pass with their skills and/or their commitment. They go through the motions of taking the course, but don't learn what they need to. Some students say, "The instructor is so nice. If I come every day and try my best, surely she/he will give me a passing grade." It won't happen. Giving you a passing grade when you haven't earned it is not only unfair to others, but it is no favor to you. You would go on to struggle in other classes.

Tutors: Tutors are available in Main 419. If you have praise or criticism for the tutors, please let me know. Please use them whenever possible. The hours for fall semester are: Monday through Saturday: 8:30 am -9:00pm.

Text: *Data Structures and Algorithm Analysis in C++, 4th Edition* by Mark Weiss. Addison-Wesley, 2013 Print ISBN-13: 978-0132847377. Be aware that there is also a third edition. You may use either one, but resale of the 3rd edition may be limited.

Objectives: The main objectives of this course are: a mastery of data structures, refining programming skills, a mastery of developing strategies for the design and evaluation of algorithms, a familiarity with algorithm analysis, a mastery of recursion, a familiarity of sorting algorithms, a familiarity of graphs, a familiarity of trees, a mastery of binary search trees, a mastery of hash tables, a mastery of priority queues, a mastery of splay trees, a mastery of merging priority queues, a familiarity of disjoint set classes, an exposure to dynamic programming, an exposure to greedy algorithms. Memorization or copying is not learning and will not be encouraged. Class discussion utilizes discovery learning and will be very different in nature from the step by step, cookbook approach of most texts. Since you will experience both the text's presentation and the derivation of the ideas in class, you will have the benefit of both teaching techniques.

Student Outcomes: C. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs. J. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

Requirements:

Programming Assignments: There are seven programming assignments during the semester. The point value of the programming assignments is not commensurate with the time involved to complete them. Programming assignments should be viewed as essential preparation for exams, rather than work that is adequately rewarded. All programs are normally written using C++. You may use any computer system you desire. Each program must be your own work (this

includes **not** allowing a tutor to write your programs). You are **not** allowed to use the STL (standard template library) unless the assignment explicitly states you can.

All programming assignments are to be submitted using Canvas. Assignments turned in after 11:59 p.m. on the date due are late. Students are responsible for turning in their programs on time. I rarely alter the due date of an assignment, and will not do so unless all students can be informed of the change at least two days before the original due date.

Use the Style Guidelines available from the class web page. Your programs are graded based on these guidelines; make sure you understand them. You will lose points for violating the standard. Unless otherwise specified, programming assignments are graded as follows:

45% Program contains no functional errors and produces correct output.

5% Format of output is pleasing and easy to understand. A person should be able to tell what the program does by just looking at the output. Put enough information in the output so this is true.

25% Efficient, well designed, extendible code.

15% Readability, good variable names.

10% Comments. Well-commented source code is often a necessity for others who will read your code. This includes explanation of variable names, functions, and descriptions of chunks of code. Note, comments should not be on every line.

As an encouragement to complete all the assignments, ten bonus points will be given to those who submit all assignments and receive at least 60% of the points possible.

Homework Quizzes: We will have a variety of online quizzes which act like written homework assignments. Answers are given through the canvas quiz system. You may work in groups of one, two, or three. They are open book and open web. Groups may change throughout the semester. Feel free to visit with me about possible answers to homework exercises. Answers should not be compared with others not in your group. Where possible, questions are automatically scored. You are allowed to repeat the exercise once to improve your score.

You will learn much more by working in a group than you will learn working by yourself. Educationally, it is a superior experience. You have to defend your answers. You get to take turns explaining and being taught. There are more of you to seek help from me, should you need it. When you do seek help, you are more confident that you have an important question as there are three of you with the same question. Thus, you don't feel "It's just me." Instead of just skipping a question you don't understand, you are able to iterate through several choices. You really work on every question. All students benefit from this thorough discussion.

Exams: There are two midterm exams (100 points each), given on October 6th and November 14th, and the final (150 points) given **Monday Dec 12th**

Exams cover material presented in class, in the book, and on the assignments. We will use the testing center for all exams. You will have a twenty four hour time period to complete the exams. The midterm exams are designed to be fifty minute exams, but you will have two hours to complete them. The final exam is designed to be a 105 minute exam, but you will have three hours to complete it. I do not give makeup exams. Please verify that you are able to take all the exams on the dates specified. Note that the exams are weighted more heavily than you may have seen in previous courses.

Because I need to be gone several days this semester, class WILL BE held on exam day to make up for other days that class will need to be canceled.

Grading:

Programming Assignments	30%
Homework Quizzes	16%
Exams	54%

The average grade for this class is designed to be a 3.1, but grades can be higher or lower depending on class performance. Grades **typically** fit the following pattern

% of points	Grade	% of points	Grade	% of points	Grade
95-100	A	90-94	A-	87-89	B+
83-86	B	80-82	B-	76-79	C+
70-75	C	65-69	C-	60-64	D+
50-59	D	<50	F		

Course Outline:

Chapter	Topic
2	Algorithm Analysis - sections 2.1, 2.2, 2.3, 2.4
4	Trees - sections 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7
5	Hashing - sections 5.1, 5.2, 5.3, 5.4, 5.5
6	Priority Queues (Heaps) - sections 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8
7	Sorting - sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7
8	The Disjoint Set Class - sections 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7
9	Graph Algorithms - sections 9.1, 9.2, 9.3, 9.4, 9.5, 9.6

Regrading: If you feel that an item has been graded incorrectly, submit a concise written summary of your concerns to me. These requests should be submitted within a week of the return of the assignment or exam.

Etiquette: My time at the university is spent in two major chunks: time teaching and time doing research. As a university professor I am expected to excel in both areas. In order for our relationship to be a happy one, you need to understand where I'm coming from.

Monday, Wednesday and Friday the majority of my time is spent with office hours, teaching, and class preparation. All my office hours are scheduled for these days.

During office hours, I try not to let one student take too much time if others are waiting. If, during my office hours, I see you have been waiting, I may ask if you have a quick question or want to listen to the explanation or may even ask the person I have been helping to do something independently while I help you. However, the decision to interrupt what I'm doing to help you is my decision.

Tardiness: When you come to class late, every person in the room is distracted by your entrance (including your professor). You miss important material. You are saying to your professor, "My time and my schedule are more important than what you have to teach me." On the job, if you don't come to work, you will be fired. If you come late, you will be reprimanded. Come to class and come on time! It is good practice for the real world.

Cell Phones: Please turn off cell phones before coming to class. When a cell phone goes off, or you are texting or reading a text message during class, it disturbs your teacher and everyone else in the class. *The penalty for any of the above is that you will bring treats for the whole class the next period!*

Time: This class takes effort. Be aware that a programming class usually demands a greater time commitment than other classes. Some of the time is unavoidable. It will take everyone a fair amount of time to get the programming assignments working correctly. However, some of the time required depends on you. Methodical programming and wise debugging can greatly reduce the programming time.

Before you complain about the amount of time you are spending on the course, ask yourself the following questions:
Do I start assignments late so class explanations are wasted because I'm not ready for the answers given?

Do I refuse to learn to use the debugger?

Do I hack at the leaves instead of getting to the root of the problem?

Do I refuse to figure out why something is happening, but try to debug totally from observing what happens when I make random changes?"

It is extremely frustrating when students complain, but are not willing to take the steps necessary to eliminate the problem. If you don't do everything in your power to make this course reasonable, I can guarantee you will be miserable! The result isn't intended, but it is inevitable.

Standards: All work in this class is graded based on the best solution rather than merely solving the problem. Sometimes students feel it is unfair for a teacher to expect the best when they are just learning, but it is important for a student to strive for the real goal. On the job, a program that "just works" is almost never adequate. How can one expect to improve or even recognized quality if it has never been demanded?

Late Work: The most common problem in this class is failure to complete the programs on time. Students are typically optimistic about the amount of time it takes to write a program, and tend to budget their time for the best possible case instead of for the average or worst case. In addition, when problems do arise, a person tends to think that she/he is the only one with such unforeseen problems and anticipates exceptions will certainly be made. Once a person gets behind with one program, it is common to be behind on many programs either because a late finish on one dictates a late start on the next or because the penalty was not sufficient to avoid similar pitfalls.

Late work creates difficulties in grading. Unless a very strict policy is enforced, chaos reigns. It is not that I am insensitive to your personal problems, but rather that I must insist that you rise above them. When the instructor grants an extension to one student, it is unfair to the other students who would have benefited from such special treatment.

All programming assignments are due at 11:59 p.m. on the date specified. However, each student is entitled to one personal emergency. You will be given a total of seven late days that you can use without penalty. They can be used on one assignment or on several different assignments. Once the late days are used, programs will be docked ten percent (of the possible) for each day late. I would strongly suggest the following practice. **On the day the assignment is due, turn in something. Then if you get more work done later, you can turn it in again. Canvas only records the last assignment turned in.**

Preparation: Preparation is necessary for learning. For this class, preparation includes attending class regularly (90% of the time), coming on time, remaining focused until class is dismissed, asking timely questions, trying problems at your seats when directed to do so, answering questions when called upon, completing homework questions, paying attention during lecture, and reading appropriate material before coming to class. You will need to bring paper and pencil. Be aware that having your laptop open often means you are diverting attention from what is going on in class. Do so at your own risk.

Because you learn more if you are involved in class discussion, I often ask for class response to a question. However, do not feel that you need to answer every question. I would like to hear from everyone in the class not the same two or three people every time. To facilitate hearing from everyone, the "three strikes rule" is implemented. After you have verbalized an answer three times in a class period, you are not allowed to answer any more questions that period. If your answer is so wonderful that you will die if it isn't expressed, tell your neighbor and let her/him share it. Along the same line, make sure the questions you ask are appropriate for the entire class. If you had a bizarre occurrence on your home computer, are wondering what you will miss when you travel to Alaska on Wednesday, or want to know how to use some advanced feature of the language, ask me after class. I don't want to spend class time on questions that are of interest to only a few or intimidate others by answering questions they are not ready for.

Academic Integrity – "The Honor System": Each student has the right and duty to pursue his or her academic experience free of dishonesty. The Honor System is designed to establish the higher level of conduct expected and required of all Utah State University students.

The Honor Pledge: To enhance the learning environment at Utah State University and to develop student academic integrity, each student agrees to the following Honor Pledge: "I pledge, on my honor, to conduct myself with the foremost level of academic integrity." A student who lives by the Honor Pledge is a student who does more than not cheat, falsify, or plagiarize. A student who lives by the Honor Pledge:

- Espouses academic integrity as an underlying and essential principle of the Utah State University community;
- Understands that each act of academic dishonesty devalues every degree that is awarded by this institution; and
- Is a welcomed and valued member of Utah State University.

Plagiarism and Cheating:

For this course, it is almost never appropriate to copy code from the book or another source. When you graduate, you will often pull code from another source, but at this stage in your development, you need to write it! When I ask you to write code that has been written by thousands of others before you, you still need to write it so you appreciate it, so you learn the associated lessons. You learn next to nothing by copying code from elsewhere. Using the standard template library is also not allowed (without specific permission from your instructor).

In English, if you are to write a paper, you are not allowed to find a good one on the web and turn it in. In CS, it is a little different, because you are encouraged not to reinvent the wheel. However, in this class, I want you to "invent the wheel." If you don't know how the wheel was built, you can't improve upon it. Learning must be done in layers. I can't teach you how to code exotic programs unless you have done the simpler things. Writing the simpler things yourself is necessary to form the correct foundation for what you need for later algorithms.

In any course, you should never use someone else's product without clearly stating where it came from. To use someone else's creation without giving them credit is cheating. Here's the approach I would try. Study the book. When an assignment is given, try to do the assignment without looking at anything. If you can't, study the book again - but before you start coding, shut the book. Then, you know that you have retained the most critical parts of the design.

The penalty for cheating is intended to have a punitive effect. If the penalty is so light, students can look at the penalty they receive and the probability of being caught and maximize their expected utility. So the thinking could be, "If I cheat, I will only get caught 5% of the time. When I'm caught it may lower my grade from a C+ to a C. However, the 95% of the time I don't get caught, I win. I save 10 hours a week in effort and get a better grade than I would have otherwise." Fill in your own parameters, but if the probability of being caught is low, unless the penalty for cheating is high, people assume they come out ahead by cheating. I do not intend to let that happen. If I don't punish cheating, I disadvantage those who are honest. I create a climate where cheating appears to be the better option. Otherwise honest people feel they have to cheat to be competitive.

Plagiarism includes knowingly "representing, by paraphrase or direct quotation, the published or unpublished work of another person as one's own in any academic exercise or activity without full and clear acknowledgment. It also includes the unacknowledged use of materials prepared by another person or agency engaged in the selling of term papers or other academic materials." The penalties for plagiarism are severe. They include warning or reprimand, grade adjustment, probation, suspension, expulsion, withholding of transcripts, denial or revocation of degrees, and referral to psychological counseling.

This course adheres to the cheating policy for courses in the Department of Computer Science posted on the bulletin board outside the CS office on the 4th floor of Old Main and posted online at <http://cs.usu.edu/htm/cheating-policy/>.

Incompletes: Students are required to complete all courses for which they are registered by the end of the semester. In some cases, a student may be unable to complete all of the coursework because of extenuating circumstances, but not due to poor performance or to retain financial aid. The term 'extenuating' circumstances includes: (1) incapacitating illness which prevents a student from attending classes for a minimum period of two weeks, (2) a death in the immediate family, (3) financial responsibilities requiring a student to alter a work schedule to secure employment, (4) change in work schedule as required by an employer, or (5) other emergencies deemed appropriate by the instructor.

Learning Aids: Lecture notes and other useful information will be available in electronic form on the class's section of

the Canvas system. Please check the class's news and notes sections on a regular basis.

The Computer Science Department is a member of the Microsoft's DreamSpark program. Through this program, students in CS courses can obtain and use a number of Microsoft's operating and software packages. If you are interesting in downloading any of this software for your use, please follow the directions found on the department's website. <http://www.cs.usu.edu/resources/elms>

ADA Statement: Students with ADA-documented physical, sensory, emotional or medical impairments may be eligible for reasonable accommodations. Veterans may also be eligible for services. All accommodations are coordinated through the Disability Resource Center (DRC) in Room 101 of the University Inn, (435)797-2444. Please contact the DRC as early in the semester as possible. Alternate format materials (Braille, large print, digital, or audio) are available with advance notice.

Class Fees: A \$60 fee is associated with this class. The monies from this fee are used to maintain lab facilities for the class, purchase software and licenses, and pay tutors. In some cases, students may have their own computing equipment, and thus feel that they do not need to use the lab. However, the lab must be maintained regardless of an individual's use of it, and thus the fee is charged to all registered for the class. If you have questions or concerns about the fee, please see the department head.

Sexual Harassment: Sexual harassment is defined by the Affirmative Action/Equal Employment Opportunity Commission as any "unwelcome sexual advances, requests for sexual favors, and other verbal or physical conduct of a sexual nature." If you feel you are a victim of sexual harassment, you may talk to or file a complaint with the Affirmative Action/Equal Employment Opportunity Office located in Old Main, Room 161, or call the AA/EEO Office at 797-1266.

Academic Freedom and Professional Responsibilities (Faculty Code): Academic freedom is the right to teach, study, discuss, investigate, discover, create, and publish freely. Academic freedom protects the rights of faculty members in teaching and of students in learning. Freedom in research is fundamental to the advancement of truth. Faculty members are entitled to full freedom in teaching, research, and creative activities, subject to the limitations imposed by professional responsibility. Faculty Code Policy #403 further defines academic freedom and professional responsibilities: USU Policies Section 403

Add Date: The last day to add this class is the 15th day of the semester. Attending this class beyond that date without being officially registered will not be approved by the Dean's Office. No assignments or tests of any kind will be graded for students whose names do not appear on the class list.

There are several reasons for this rule. Students who attend classes without registering have an unfair advantage over those who are registered. The unregistered student can choose not to register if their academic performance is poor. There will be no record that the student "withdrew." The registered student must drop the course along with the ramifications of small or no refund in tuition and a possible "W" on his/her transcript.

The university does not receive the headcount credit from the State for students who add any class after the 15th day. We lose a significant amount in support funding for those students not registered prior to day 15. Students who attend classes without registering/paying are utilizing campus resources even though they have not paid tuition and fees. Students who are attending classes but not enrolled in them are not subject to the Student Code of Conduct. Thus, we are asked not to allow students to "sit in" on classes for which they are not enrolled.

Drop Date: The last day to drop classes (with the "W" notation on transcript) is Oct 31st. If you are considering dropping, you should talk to me about the desirability of such a decision. Sometimes students drop when they would actually do well with a different strategy. Other times students who should drop, do not. It is wise to consult your instructor in your decision as she may have a different reading on your situation.