



Data Structures

CSE 2341 – Spring 2019

Introduction:

Welcome to Data Structures! During this semester, we will delve into the world of object oriented C++, data organization and access, and associated software development topics. A major aspect of computer science is finding an efficient solution to a given problem that can be implemented on a computer. We will explore some of the more intermediate and advanced parts of the C++ language and how they can be leveraged in construction and implementing various data structures and algorithms to operate on those data structures. We will also begin to explore how data organization and access can enhance or inhibit the efficiency of a solution to a problem.

As a similar saying goes, programming languages are to computer science what hammers are to carpentry. Programming is a tool that we use to communicate with a computer and express our solutions to problems. It is important to know how to use this tool efficiently to solve complex problems. To get the most out of this class and ultimately to be successful, you must be willing to **work hard**.

Instructor Information:

Mark Fontenot, Ph.D.
Office: Caruth 441
mfonten@lyle.smu.edu
214.768.2854 (please use email for routine contact)
<http://lyle.smu.edu/~mfonten> <http://markfontenot.net>
Office Hours: Open Door Policy

Course Information:

Lecture:

TTh: 9:30-10:50 pm Junkins 113

Lab Sections:

N11:	W 8:00 - 9:50	Embrey 123
N12:	Th 3:00 - 4:50	Embrey 123
N13:	W 3:00 - 4:50	Junkins 102
N14:	Tu 5:00 - 6:50	Caruth 485
N15:	Tu 3:00 - 4:50	Embrey 123
N16:	W 4:00 - 5:50	Caruth 485

Catalog Description:

Emphasizes the object-oriented implementation of data structures, including linked lists, stacks, queues, sets, and binary trees. The course covers object-oriented software engineering strategies and approaches to programming in interdisciplinary teams.

Where to Find Info:

Course Management:	We'll be using Canvas. You can access it via http://smu.instructure.com
Document Location:	markfontenot.net
Quick Communication & Questions:	piazza.com and Slack
Project submissions:	github.com

Recommended Textbook:

The material covered in this course has been refined over the past six or seven years. To date, I have not found a textbook that covers all the topics that we will encounter this semester. Additionally, I believe students must learn to teach themselves and to find relevant information on a topic. Every bit (no pun intended) of the content knowledge that we will cover is somewhere on the web. Each topic is likely covered hundreds or thousands of times there. I encourage you to use your web savvy in this course.

There is very little chance that you'll find answers to homework assignments or programming projects from this class online. What you will be able to find online is knowledge that will aid you in solving the problems you encounter. So, channel your inner "Googl-er" or "Bing-er" or whatever other search engine that you find palatable. Read about the topics we cover. Read several different descriptions or explanations of each topic. Read until you begin to understand. Watch YouTube videos. You even have my explicit permission to read Wikipedia for this course; there are some great explanations of challenging topics contained within its digital walls.

Here are a few e-book/references that I will sometimes refer to in class (and most are freely available)

- Shaffer, C. *Data Structures and Algorithm Analysis, C++ version*, Edition 3.2, Dover publications. Available in PDF at <http://people.cs.vt.edu/~shaffer/Book>

Note: Yes, folks, this one is indeed free. Dr. Shaffer wrote this book, and I think it is great. His explanations of theory are fantastic, and his code examples are quite good as well. At times I will refer you to different parts of this book to read up on topics. It is this text that I probably follow most closely.

- Kempe, David. *Class Notes for CSCI 104: Data Structures and Object-Oriented Design*. July 13, 2016. Available in PDF at <http://www-bcf.usc.edu/~dkempe/teaching/DataStructures.pdf>

These are the lecture notes of Professor Kempe from USC for their Data Structures course. Again - they are very well written and clear. There are multiple editions of this around the web. You don't have to find one that is precisely dated as above.

OR

- Carrano, F. *Data Abstraction and Problem Solving with C++, 7th ed.* Pearson. ISBN: 9780134463971

Note: If you find a cheaper one that is one or two editions old, that's fine. Dr. Carrano's explanations of conceptual topics are really great, and the code examples are not too bad. I will, however, not explicitly require you to ever read any topics from this text specifically. Think of this as a reference text.

Prerequisite

C- or better in CSE 1342 or equivalent or permission of instructor.

Learning Outcomes:

1. Ability to properly handle dynamic memory management tasks in c++
2. Ability to compare data structures and their algorithms analytically and experimentally
3. Ability to implement a range of data structures including but not limited to linked lists, stacks, queues, binary search trees, hash tables, and heaps
4. Ability to compare and contrast various comparison-based sorting algorithms
5. Ability to decompose and solve large problems
6. Ability to choose appropriate data structures to solve a problem
7. Ability to design, implement, test, and debug classes in C++

Method of Evaluation:

The final course grade in CSE 2341 will be based upon performance of various deliverables such as implementation projects and homework assignments. Additionally, your attendance and active participation in lecture and lab will be reflected in your final grade. The percentage breakdown is as follows:

Evaluation Tool	Percentage
Exams	25%
Programming Assignments	30%
Semester Project	20%
Homework Assignments and Quizzes	15%
Attendance and Active Participation in Lecture and Lab	10%

There will be a midterm and final. I believe in **learning by doing**, and there is quite a bit of research to support learning-by-doing as valid and relevant. We will routinely jump back and forth between the abstract/theoretical and the applied domains. You will complete a number of programming projects that will require substantial time investments. The material covered in this course, however, is often used in technical interviews for internships and full-time positions. So it is vitally important that you master it.

The midterm or final for this course may take the form of a **mock technical interview** or a **written exam**.

The due dates for all assignments will be explicitly stated on each assignment and will be available on the course website. Final grades in this course are determined as follows:

90 - 100 : A		
87 - 89 : B+	83 - 86 : B	80 - 82 : B-
77 - 79 : C+	73 - 76 : C	70 - 72 : C-
67 - 69 : D+	63 - 66 : D	60 - 62 : D-
0 - 60 : F		

Additional grade stipulations and notes:

- In order to get a grade above a D+ in the course, you must:
 - maintain a passing average (>60%) on all regular programming assignments, **and**
 - obtain a passing score on the final course project.

Homework Assignments:

Homework assignments are assigned throughout the semester. The assignments will be available via Canvas and/or will be handed out in class. Each will clearly state the due date and method of submission. It is important to take the homework sets seriously. Questions posed in the final mock interview or written final will be more consistent with homework problems than with programming projects.

Programming Projects:

Nearly every week, you'll be working on some aspect of a programming assignment in and out of your lab period. **You will not be able to finish an entire assignment during lab time;** if you do, you should question whether or not you actually fulfilled the requirements. Further, your lab period is not the time to start your work for the week. You should work on your assignments before lab and use lab as time to get clarifications and explanations as needed from the TAs. Each lab packet will clearly state the due dates for each of the deliverables.

Submission Deadlines:

Rather than penalize late submissions, you will be rewarded for early submissions. Programming projects (except for the final semester project) will be eligible for 10% extra credit for submitting the project at least 48 hours early. Recognizing, that sometimes things happen at the last minute that are beyond our control, each student will get one (1) two-day (2-day) "free" extension - no questions asked. This means you can submit the project (except for the final semester project) up to 48 hours late. You need to alert Dr. Fontenot before the end of that 48 hour period that you're using your one free late submission.

General Requirements for all Programming Projects:

All projects will be reviewed, evaluated, and graded using the virtual environment (virtual machine) provided by the course staff. Your submission must conform to the specifications in each project handout. Each submission is required to use the qmake build system to compile and run your projects. The appropriate files for this build system are automatically included in the project directory when you create a new project with Qt Creator, the IDE used for this class.

Project submissions that do not build/compile using the above stated build system and/or any modifications to the build system as stated in each individual project handout will not be given any credit. No matter how much time you spend writing code, if it doesn't build/compile, you will receive NO credit.

We will use GitHub throughout the semester as a code versioning and backup tool. If used as instructed (and as GitHub is intended), you will always have a safe and secure backup of your source code. Therefore, I will entertain no excuses related to computer hardware failure for not submitting projects or assignments on time.

Communication, Attendance, and Grading:

Because of the nature of this class, attendance of and participation in lecture and lab is very important. I expect you to attend all classes and lab sessions. If you are absent from class, it is your responsibility to make arrangements with me to make up any work missed or to ensure that assignments are submitted on time or early. **Except as stated above in Programming Projects section, no late work will be accepted.** Any assignments that will be missed (including those due to university-sanctioned events) must be completed **before the due date**. This includes projects, exams, and homework assignments.

Note that ten percent (10%) of the semester grade is based upon class attendance and active participation. This attendance score will be determined based attendance and active participation in both lecture and lab.

Attendance in lecture will be randomly taken. Lab attendance will be taken weekly. After three missed lab or lecture classes, on point will be deducted from your final course average for each missed class or lab missed.

Any issues with grades assigned on projects, homeworks, quizzes, exams or any other graded materials must be brought to Dr. Fontenot's attention within 48 hours of the grades being posted on Canvas, or they will not be entertained. This should be done via email.

A college course is taught by a professor and attended by a student. So, the educational engagement is between the two of us, and informed by the course staff of graduate and undergraduate teaching assistants. This educational engagement does not include your parents, guardians, or others in any way. Therefore, I will only be communicating with you. I will not be communicating with your parents or guardians in any way, regardless of the presence of a FERPA release that you have issued.

Outside Work:

This course will require a substantial amount of work outside of class. The best way to learn this material is to sit down, get comfortable, and just do it. There will be several programming assignments and projects that will require a great deal of time on your part. Some of these projects may be collaborative in nature. It is reasonable to expect that you may spend 10 or more hours per week outside of lecture/lab working on programming projects for this class.

Here are some soft-skills that we hope to help you hone over the course of the semester:

- Time Management (i.e. not waiting until the last minute to begin work on projects and assignments)
- Work smart by using your time wisely and using tools related to the course material effectively
- Large-scale problem solving (i.e. planning and designing before coding)

Academic Ethics and Collaboration:

I take academic ethics very seriously. Each of you should have read the SMU Honor Code by this point in your academic career at SMU. What you submit must be your own intellectual work product, except where collaboration with other students is explicitly allowed. In addition, the metadata associated with your work product (file timestamps, commit logs from GitHub, etc.) are also part of your work and will be treated as such. Any modification of these metadata done to intentionally mislead one of the course staff members will be treated as a cheating offense.

You are expected to create, edit and print your own assignments and take tests without outside assistance. All work is expected to be your own. In particular:

- You should never look at or review another person's work for any given assignment. This includes looking at papers, solutions, or computer screens where another student's work is displayed. This also includes solutions from previous semesters.
- You should never give or receive solutions/answers to any questions or projects or any parts or questions or projects. This includes but is not limited to source code, design documents, homework, etc.
- In the header comments of your source code, you must reference online sources you consulted to complete the project. This is especially important if you've "quoted" source code.

If you collaborate on any assignment for any reason unless specifically permitted by the instructor, you will receive an F in the course and may be brought in front of the SMU Honor Council. The presumption should be that collaboration is not permitted and collaboration should only take place subsequent to explicit instructions by the course instructor. It is your responsibility to know and understand the University's Honor Code and the expectations for collaboration in this course. Dr. Fontenot reserves the right to impose less severe penalties at his sole discretion.

Words of wisdom: **The penalty for academic dishonesty will always be more severe than simply not submitting the assignment.** Not turning in an assignment will result in a zero on that assignment. Engaging in academic dishonesty related to that assignment will likely result in an F in the entire course and/or referral to the honor council.

So, what is explicitly permissible? I will be the first to agree that sometimes we learn better when we are talking through things. When a project is assigned, you are certainly allowed to 'whiteboard' design ideas, high level plans, etc. But they should never get to the level of code. If you have questions about what is allowed or not allowed, please just ask Dr. Fontenot.

Disability Accommodations:

Students needing academic accommodations for a disability must first register with Disability Accommodations & Success Strategies (DASS). Students can call 214-768-1470 or visit <http://www.smu.edu/Provost/ALEC/DASS> to begin the process. Once registered, students should then schedule an appointment with the professor as early in the semester as possible, present a DASS Accommodation Letter, and make appropriate arrangements. Please note that accommodations are not retroactive and require advance notice to implement.

Religious Observance:

Religiously observant students wishing to be absent on holidays that require missing class should notify their professors in writing at the beginning of the semester, and should discuss with them, in advance, acceptable ways of making up any work missed because of the absence. (See University Policy No. 1.9.)

Excused Absences for University Extracurricular Activities:

Students participating in an officially sanctioned, scheduled University extracurricular activity will be given the opportunity to make up class assignments or other graded assignments missed as a result of their participation. It is the responsibility of the student to make arrangements with the instructor prior to any missed scheduled examination or other missed assignment for making up the work. (University Undergraduate Catalog)

Conduct Expectations:

Students shall observe the following:

- Using a laptop in class is a privilege; it is not a right. When laptops distract you or other students in the class you will no longer be allowed to use them. I reserve the right to ask to see your screen and class notes taken during the class to determine if you are using your laptop appropriately.
- Cell phones should be in vibrate/silent mode at all times during lecture. If your cell phone does not have this feature, then it should be turned off. DO NOT answer your cell phone during class.
- Sending text messages, surfing the Internet, playing games, reading the newspaper, etc. is considered disrespectful and will be dealt with accordingly. You are not only disrespecting the instructor, you're also distracting students around you whether you think so or not.
- Students should remain quiet during lecture so as to avoid disturbing other students.
- Yawning out loud is disrespectful!

Tentative Schedule of Topics

This schedule is tentative and subject to modification throughout the semester.

Week Of:	Topics:
Jan 21	Getting Started, C++ Review
Jan 28	C++ OOP & Memory Management
Feb 4	C++ OOP & Memory Management
Feb 11	Recursion & Algorithm Analysis
Feb 18	Sorting
Feb 25	Linear Data Structures
Mar 4	Exam Review Thursday - Midterm.
Mar 11	Spring Break
Mar 18	Trees
Mar 25	Balanced Trees
Apr 1	Hash Tables
Apr 8	Project Kickoff on Tuesday Priority Queues
Apr 15	Sets
Apr 22	Monday Project Check-ins Graphs
Apr 29	Graphs
May 6	Project Demos on May 6

There will be a final exam for the course, and it has historically been an oral final. Before you make travel plans for the end of the semester, check with Prof. Fontenot about the dates for the final.