SMU | Bobby B. Lyle School of Engineering

Data Structures

CSE 2341 – Fall 2015

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:

Professor Mark Fontenot
Caruth 441
mfonten@lyle.smu.edu
214.768.2854
http://lyle.smu.edu/~mfonten
Office Hours: Mondays, Wednesdays 3-4pm
(You are welcome to stop by my office anytime my door is open. These are just my "official" office hours. I know these times won't work for everyone.)

Course Information:

Lecture: MWF: 2:00 – 2:50 am Junkins 113
Enrollment in the lecture requires enrollment in one of the associated lab sections for CSE 2341

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: courses.smu.edu

Document Location: markfontenot.net
Piazza for Questions: piazza.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 on the web. 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 You Tube 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 is a e-book that I will sometimes reference and that is 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.

OR

- Carrano, F. *Data Abstraction and Problem Solving with C++, 5th ed.* Pearson. ISBN: 9780321433329

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.

Pre-requisite

C- or better in CSE 1342 or equivalent.

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
Midterm And Final Exam	20%
Programming Assignments	25%
Semester Project	25%
Homework Assignments and quizzes	20%
Attendance and Active Participation	10%

There will be a midterm and final. I believe in the adage **learn by doing**, and there is quite a bit of research to support learning-by-doing as a valid and relevant method of learning. The material covered in this course, however, is often used in technical interviews for internships and full-time positions. Therefore, the 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 (>70%) 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 Blackboard 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. But, 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.

Programming projects must be submitted per assignment instructions before the due date. Only the professor can give an extension to this due date (a typical extension is 48 hours). Any extensions will incur a minimum 10% penalty on the grade for that project. The professor reserves the right to assign a more severe penalty at his sole discretion.

Source code that does not compile will not be given any credit. Yes, that's right. No matter how much time you spend writing code, if it doesn't compile, you will receive NO credit.

Attendance Policy:

Because of the nature of this class, attendance of and participation in lecture 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. Late assignments will not be accepted except in extreme cases. 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 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.

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 to 15 hours per week on this course outside of lecture/lab with some weeks peaking at 20 or more hours.

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:

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 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. The instructor of this course reserves the right to impose less severe penalties as he sees fit.

Words of wisdom: **The penalty for academic dishonesty will always be worse than not turning in 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.

Disability Accommodations:

Students needing academic accommodations for a disability must first contact Disability Accommodations & Success Strategies (DASS) at 214-768-1470 or www.smu.edu/alec/dass.asp to verify the disability and to establish eligibility for accommodations. They should then schedule an appointment with the professor to make appropriate arrangements.

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 should observe the following:

- 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:**
Aug 24	Intro to class; Memory Management
Aug 31	Memory Management
Sep 7	Advanced C++ Topics (Monday = Labor Day)
Sep 14	Advanced C++ Topics & Recursion
Sep 21	Algorithm Analysis
Sep 28	Sorting
Oct 5	Lists, stacks, and queues
Oct 12	Trees (Monday = Fall Break)
Oct 19	Trees
Oct 26	Trees
Nov 2	Trees (Nov 6 = Last day to drop class)
Nov 9	Priority Queues/Heaps
Nov 16	Hashing
Nov 23	Hashing
Nov 30	Graphs
Dec 7	Monday Class Only - Project Demos.