

Introduction

CS-590 uses C++ to introduce students to intermediate data structures and algorithms. **Strong experience in introductory programming and basic data structures is expected as a prerequisite for this course. Additionally, C++ experience or a willingness to rapidly learn C++ with limited assistance is necessary to complete this course.**

The course textbook is [A Practical Introduction to Data Structures and Algorithm Analysis, Third Edition \(C++\)](#) by *Clifford A. Shaffer*.

Other course materials will be posted on Canvas from time to time.

Every few weeks, a live "office hours" will be held during which course material will be reviewed and students may sign on to ask questions. The dates and times of these sessions will be posted at least one week in advance of when they occur.

Outcomes

These are the official approved outcomes for this course:

- **Complexity** – Explain the meaning of big-O, Theta, and Omega notations. Calculate the asymptotic running time of standard algorithms, and use it to compare efficiency.
- **Master Theorem** – Use the Master Theorem to prove asymptotic assumptions
- **Sorting** - Compare and analyze basic and advanced sorting algorithms.
- **Trees** - Implement advanced search trees such as Binary Search Trees, and Red-Black Trees.
- **Graphs** - Implement standard algorithms using graphs and weighted graphs in C++ (e.g., DFS, BFS, MST, topological sort).
- **Shortest Paths** – Implement standard algorithms to solve the shortest path finding problem. (Dijkstra, Bellman-Ford, Floyd-Warshall)
- **Algorithmic Design** - Apply standard algorithm design techniques such as the greedy technique, dynamic programming, hashing, and space/time trade-offs.

Course Breakdown

Spring 2019 is the first semester that this "new" version of CS-590 will run. As such, all graded works are not known yet at the beginning of the semester.

There will be a mixture of assignments and quizzes that you will complete that will be worth a varying amount of points to earn the final grade. **There will be no midterm or final exam.** However, there will be a major programming assignment worth a substantial number of points during the finals period of the semester.

Given that the course material will change dramatically this semester, it is important that you communicate via the discussion board if assignments or quizzes are too difficult for the time allotted so

that adjustments to the course schedule can be made.

Assignments

Assignments must be completed individually – no groups.

Assignments must be submitted on time in order to receive full credit. Assignments will have a -10 deduction applied to them for each day that the assignment is late, except that assignments that are less than 6 hours late receive a -5 deduction. You may resubmit as many times as you want **before** you receive a grade. However, your most recent submission will always be the one graded. Grading may commence at any time after you submit, even if before the deadline; if you do not want your submission graded, you **MUST** note that in the comments ASAP so that it is not graded.

Assignments may not be resubmitted for any reason.

Assignments are first graded automatically by a computer script. If your program does not match the exact input/output expected by the script, you may get a significantly lower grade than you expected.

The following requirements for assignments must be strictly followed:

1. Assignments will be graded on Ubuntu Linux 18.04.1 LTS with package *build-essential* and using the *g++* compiler. *g++* reports the following version information when the "*g++ --version*" command is run:

```
g++ (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Your code must compile on this grading environment. It is recommended that you update the version of this package immediately prior to submitting your assignment.

2. Assignments must be zipped up into a *.tgz* "tarball" file where the root of your project is the root of the tarball. The easiest way to create this is to execute the following command in the root of your project:
*tar czvf assignment.tgz **
3. Assignments must be able to be compiled by executing the following command:
make
4. If an assignments has a requirement that it must be executable, your assignment must be able to be run by executing the following command:
./assignment
5. Each code file in an assignments must contain the academic honesty comment text found at the bottom of this syllabus, verbatim with no modifications whatsoever.

If any of the above rules are not followed, you will receive an automatic 0 on your assignment and no resubmission will be possible. The time to get assistance with the requirements is before the due date, not after you have submitted and received a 0.

All extensions must be requested before the due date of an assignment. After-the-fact extensions will only be considered in cases where it would have been impossible to request an extension before the due date.

Quizzes

Several quizzes may be given out throughout the semester. Quizzes will be available for one week from the date assigned, and will be announced with at least 5 days notice. Once you start a quiz, you will have a limited amount of time to complete it. The quiz will automatically prevent submission once the time has elapsed, and you will be unable to submit.

Quizzes are individual work. Some quizzes may be open book, and may be closed book. For closed book quizzes, Canvas uses a product called Lockdown browser in conjunction with your computer's webcam to verify that work is completed on an individual basis and that no references are used. It is imperative that you ensure your computer is compatible with the Lockdown browser prior to attempting a quiz.

Class Communication

The primary mechanism for communication in this course is this Canvas's discussion board. You are responsible for being aware of clarifications to the assignment, etc., that are posted on the discussion board. For general questions regarding the assignments, programming questions, or questions about the content or lectures, you *must* post the question to the discussion board so that all other students may benefit from the response.

For specific questions about you as an individual in the course such as your grade, requests for extensions, etc., you can contact the professor directly. **You must use email instead of Canvas Mail.** I respond to email regularly, but not to Canvas Mail. If you do not post or email in the correct location, please expect a polite but firm reply requesting that you resend your message.

Students must make contact within 7 days of receiving a grade if they want to contest their grade for an assignment, lab, or exam. After 7 days, all grades are final, even if an error was made in grading.

Academic Honesty

Academic honesty is taken seriously in this course. Manual review of work submitted and automated tooling are both used to ensure that assignments are your own work.

It is not acceptable to use code or references without prior approval from the professor on the discussion board. This includes code from the textbook or lecture notes, though you may use those as references. The only exception to this policy is that you may use any "starter code" provided by the assignment. For

clarification, a *reference* is any electronic media, print media, or other person who assists with the assignment in any way.

You may not share your work with any other students or permit them to see your solutions. You may not post your work on any publicly available web site. If you do share your work or post it online, and another student plagiarizes your work, you will be subject to the same penalty for academic dishonesty as the students who use your work. **Students repeating the course may not use their previous semesters' work towards the current semester.**

You must include the following text in every code file you submit to receive a grade other than 0 for an assignment, and you must abide by the rules within:

```
// On my honor:  
//  
// - I have not used code obtained from another student,  
// or any other unauthorized source, either modified or unmodified.  
//  
// - All code and documentation used in my program is either my original  
// work, or was derived, by me, from the source code provided by the assignment.  
//  
// - I have not discussed coding details about this project with anyone  
// other than my instructor, teaching assistants assigned to this  
// course, except via the message board for the course. I understand that I  
// may discuss the concepts of this program with other students, and that  
// another student may help me debug my program so long as neither of us  
// writes anything during the discussion or modifies any computer file  
// during the discussion. I have violated neither the spirit nor letter of this restriction.  
//
```

If you violate any part of the academic honesty policy, you will likely receive an F for the course grade with no warning. At the very minimum, will you will receive a penalty of a **0** grade for the assignment. There are **no exceptions** to this minimum penalty policy and it will be strictly enforced no matter the personal circumstances.