CMSC 262
Data Structures and Algorithms in Application
Fall 2021
http://marmorstein.org/~robert/Fall2021/cs262.html

Lecture (Rotunda 356): 4:00pm – 5:15pm MW
Instructor: Robert Marmorstein, 395-2185, marmorsteinrm@longwood.edu
Office Hours: 3:00pm – 4:00pm MTWF (Rotunda 329)

I am also available by appointment. My schedule is posted near my office door. To make an appointment, please check the schedule to see which times I am free, then contact me by e-mail and list some possible times we could meet. In general, I need at least 24 hours of notice to schedule an appointment.

Communications Policy:

The best way to get in touch with me is to use **Slack**. Slack is a chat utility with clients for mobile devices and desktop computers. It will allow you to easily send me code snippets. Also, since I get notifications when a Slack message comes in, I am more likely to reply to your message quickly than if you send me e-mail. Slack also allows me to easily set up a Zoom meeting (or Google Hangout) if we need to video chat.

When you send me a Slack message, I instantly get a notification on my computer, tablet, and phone. Typically, I will reply to Slack messages within 24 hours (often sooner) on weekdays and sometimes even on weekends. I am much slower at replying to e-mail (since I do not get a notification and have to explicitly check it). Typically, you can expect a reply to an e-mail within 48 hours (longer on weekends).

Slack is also a good way to communicate with other members of the class. You will be invited to a public #cmsc-262 channel in which you can discuss the projects and other course topics with other students in the class. Feel free to ask for help on this channel, but please stick to general questions rather than posting code.

You can also reach out to me by e-mail to marmorsteinrm@longwood.edu. However, please do not send me large files by e-mail. They take up space toward my limited quota on the mail server and cause me all sorts of headaches. **E-mail messages containing large files will be deleted unread**.

Asking for help

If you are asking for help with a project or homework problem, you can send me a direct message through Slack. You should attach your code or your work to a **Slack** message so that I can see where you are at. You should do this by using the "paperclip" icon to attach the file directly to your message or by copy/pasting the particular snippet of code you are working on to the body of the message.

Please do NOT attach screenshots or pictures taken on your phone. They are hard to read and do not allow me to compile your code without retyping it.

One last suggestion: don't "ask to ask". I am delighted to answer questions about the projects and homework assignments and you should feel free to ask questions at any time. Asking me whether you can ask a question wastes my time and yours.

Course Description: Introduces many of the classic advanced data structures and algorithms in the context of a survey of important applied fields of computer science. Topics include artificial intelligence, relational databases, and human-computer interaction. 3 credits.

Prerequisite: ***CMSC 162, MATH 175 recommended.***

Textbook: The textbook for this class is "Data Structures and Algorithms in C++", 4th edition, by Mark Allen Weiss, Pearson, ISBN: 976-0-13-284737-7.

In addition, we will be using some articles that I have placed on reserve in the library. See the course web site for details.

Course Student Learning Outcomes: By the end of the course, the successful student will be able to:

– build an interface for an application which satisfies the principles of universal design
– implement advanced tree structures such as balanced binary search trees
– use path-finding algorithms, indexing, or indirection to search multi-dimensional data
– create an application which uses relational databases to support business logic

Course Structure and Student Expectations: This class will be heavily project-driven. In addition to the three hours you spend in class each week, you should expect to spend roughly six hours outside of class working on projects, reading from the textbook, and completing homework exercises and quizzes.

Projects: There will be four major projects in this course. Each project is designed to be finished over the course of three weeks. Each of the major projects will be split into multiple graded parts – a design document, a prototype, and a final project. The projects will be group assignments and you may form groups of up to three students. I will let you choose your own groups. Since the projects will not build on each other, you may switch groups between projects. You will be graded both on the success of your group and your individual contribution to the project.

Tests: The only exam in this course will be the final exam. It will be a cumulative exam covering topics from lecture, projects, and the textbook. It will be completed remotely and submitted through Canvas. It will be a closed-book, closed-notes test.

Quizzes and Homework Problems: Each week, you will have assigned reading which you should complete before class the following Monday. There will be an open-book, open-notes online quiz due each Monday which will count toward your homework grade.

Course Requirements: Your grade will be determined by your performance on the final exam (25% of your grade), course projects (50%) and homework/quizzes (25%). See the course schedule below for due dates.

Attendance:

This class is heavily lecture-driven and will require your regular attendance. Unless you are unable to come to class for health reasons (or an official school sponsored event), I will expect you to attend in person.

In accordance with Longwood policy, missing more than 10% of scheduled class time will result in loss of one letter grade. Students who miss more than 25% of classes, whether excused or not, may at my discretion receive an F for the course.

Absences for school events, illness, or other exceptional circumstances may be excused if you contact me by e-mail prior to or within 12 hours of the missed class. To seek such an accommodation, your e-mail should explain the reason for your absence in sufficient detail that I can determine whether the absence should be excused.

In the event you are sick, it is your responsibility to make up any missed work and get notes over the missing content.

In general, I will not stream lectures for students who miss for illness. However, if you may need to miss class for an extended period of time (such as for COVID quarantine), I may be able to adapt lectures to a format suitable for streaming. To request this accommodation, send me documentation of a legitimate health problem at least 48 hours in advance.

Grading Policy:

Late work will not be accepted unless you have a medical condition or serious emergency which prevents you from completing the assignment on time. In such circumstances, you do not need a doctor's note, but you must contact me by e-mail before the assignment is due (or within 12 hours of the deadline) to explain the circumstances and arrange to make up the work. Such exceptions are granted very rarely. Technical problems involving the use of the submit system, your computer, or lab resources are not valid reasons to submit work late (that is what slip days are for).

Final letter grades will use the following scale:

	A: 91 – 100%	A-: 90%
B+: 89%	B: 81 – 88%	B-: 80%
C+: 79%	C: 71 – 78%	C-: 70%
D+: 69%	D: 64 – 68%	F: 0 – 63%

(Note: there is no grade of D- in this course)

Mental Health, Disabilities, and Sexual Assault:

I follow Longwood's campus policies on Mental Health, Disability Accommodation, and Mandatory Reporting of Sexual Assault and Other Crimes, Please see http://www.longwood.edu/academicaffairs/syllabus-statements/ for the complete policy.

Face Coverings and Intellectual Property:

Please also see http://www.longwood.edu/academicaffairs/syllabus-statements/ for the university policies on wearing masks to class and a statement on intellectual property. Both of these policies apply to this class.

Cell Phones and Laptops:

Cell phones, music players, and laptops must be turned off and put away during lecture and class discussions unless I have specifically requested that you use them. Violations of this policy will be considered an unexcused absence and may also affect your homework or participation grades.

Food and Drink:

Please do not eat in class (it distracts me and the other students). You may bring water or other non-alcoholic beverages to class. I occasionally make exceptions to this rule for students who would otherwise miss lunch or have medical needs that require them to eat in class. If you feel that you need such an exception, you MUST make arrangements with me before you bring food to class (preferably by e-mail). Violations of this policy will be considered an unexcused absence and may also affect your homework or participation grades.

Honor Code:

I take the honor code very seriously. I encourage you to take advantage of the freedom it gives you to collaborate with other students and to use print and Internet resources to better understand the material.

Because it is possible to abuse these resources in a way that actually hinders you from learning or disadvantages other students, I have established some guidelines for their use that you MUST follow.

Please read these rules carefully. It is your responsibility to know them and follow them.

Exams and quizzes are to be completed entirely on your own. Exams will be closed book/closed notes tests on which you may receive no external help and may use no resources other than your brain and a writing instrument (unless explicitly stated in class).

On homework assignments and projects, you may discuss your work with other students subject to these restrictions:

1. Turn in only your own work

The work you submit should, in general, be your own original work or material which I have provided and you have suitably modified. You **MAY** discuss problems with others in a general way. You **MAY** assist other students (or get assistance) with simple problems like syntax errors, but you **MAY NOT** copy solutions or large blocks of code from each other. You **MAY** use web sites, books, and other resources as references, but you should not use large blocks of code from these sites, either.

The definition of "large" in this case varies somewhat based on context, but a good guideline is that while copying one to three lines of code is usually okay, copying a complete function, class, or file is usually too much.

The purpose of this rule is to ensure that you understand the code or answers you are submitting. If you don't think you could explain your work to me without help or looking at a book or web page, you probably should not submit it.

2. Give proper attribution

If you **DO** get help or use an online resource to complete a project, you **MUST** give credit to your source. Taking credit for someone else's work is a form of intellectual theft called plagiarism. To cite a source for homework assignments, you can simply add a note in the margin next to the answer on which you received help. In projects, you should comments to your code. A simple comment like

```
/* Based on http://codewarrior.com */
```

or

```
// Susan helped me with this step
```

is fine.

The comment should go **directly above the line or lines on which you received help** to make it clear which parts of your program are original and which are derived from other sources, NOT at the top of the file or in a separate document.

You do not need to cite help you have received directly from me or from the textbook.

3. Do not copy code electronically

Any work you turn in should be work which YOU have typed or hand-written unless I have explicitly instructed you to download or copy/paste it. Typing in blocks of code helps you retain what you have learned better and acts as a safeguard for rule two.

You **MAY NOT** share code with other students using flash drives, cell phones, e-mail, web sites, floppies, CDs, or any other electronic storage or communication device unless you are both assigned to the same group for a project. You **MAY NOT** print out copies of your code to share with other students (personal copies or copies to bring to office hours are fine as

long as you don't leave them lying around the lab). You **MAY** copy or download any code I have posted to Slack or the course web site for your use.

4. You are responsible for securing your code.

Helping other students to cheat is also cheating. Furthermore, it is your responsibility to make sure that other students do not use your work to cheat. Be careful with who you let access your computer and report any missing files, flash drives, or other devices to me promptly.

Infractions of these policies will be dealt with harshly under the Longwood Honor Code. Any student convicted of an honor offense involving this class will automatically receive a final course grade of **F** in addition to any penalties imposed by the Honor Board. You should consider all work in this class to be pledged work, whether or not the pledge appears on the assignment.

Tentative Course Schedule:

Week 1: Aug. 23 – 25	Introduction, Review of Data Structures and Abstract Data Types **Read Chapter 1**
Aug. 31	**Last day of Add/Drop (by 5pm)**
Week 2: Aug. 30 – Sept. 1	Human-Computer Interaction: Principles of Design, Affordances, Accessibility Using Qt for Graphical Interface Design, Signals and Slots **Read Supplemental Readings 1 and 2 (details on course web site)** Project 1: Designing a Drawing program (Design Document due)
Week 3: Sept. 6 – 8	
Sept 6	**LABOR DAY HOLIDAY: NO CLASS**
	Using QGraphicsScene and QGraphicsView The Model Human Processor, Power Law of Practice **Read Supplemental Reading 3 (See Course Web Site)**
	Project 1: Designing a Drawing program (Prototype due)
Week 4: Sept. 13 – 15	Mathematical Review: Logarithms and Asymptotic Analysis **Read Chapters 2 through section 2.5 and Chapter 3, Sections 3.1 through 3.10** Project 1: Designing a drawing program (Project due)
Week 5: Sept. 20 – 22	Trees: Binary Trees, Binary Search Trees **Read Chapter 5** Project 2: Implementation and Applications of Trees (Design Document due)
Week 6: Sept. 27 – 29	Trees: Balanced Binary Search Trees and Spatial Trees **Read Chapter 13** Project 2: Implementation and Applications of Trees (Prototype due)
Week 7: Oct. 4 – 6	Trees: Non-binary Trees Project 2: Implementation and Applications of Trees (Project due) **Read Chapter 6**
Week 8: Oct. 11 – 13	Internal and External Sorting **Read Chapters 7 and 8** Project 3: Relational Databases (Design Document due)
Week 9: Oct. 18 – 20	Searching, Hashing, Divide and Conquer **Read Chapter 9** Project 3: Relational Databases (Prototype due)
Week 10: Oct. 25 – 27	Indexing and B-Trees

	Relational Databases **Read Chapter 10** Project 3: Relational Databases (Project due)
Week 11: Nov. 1 – 3	Graphs: Definitions and Traversals Spanning Trees, Prim's Algorithm, and Kruskal's Algorithm **Read Chapter 11** Project 4: Path-finding Algorithm (Design Document due)
Week 12: Nov. 8 – 10	Artificial Intelligence, Heuristics, and Search Spaces Project 4: Path-finding Algorithm (Prototype Due)
Week 13: Nov. 15 – 17	Dynamic Programming **Read Chapter 16** Project 4: Path-finding Algorithm (Project Due)
Nov. 17	**SYMPOSIUM ON THE COMMON GOOD: NO CLASS**
Week 14: Nov. 22	Catchup and Review
Nov. 24 – 26	**THANKSGIVING BREAK: NO CLASS**
Week 14: Nov. 29 – Dec 1	Catchup and Review
Dec. 8	**Final Exam (Wednesday, 3:00-5:30pm)**