CMSC 162
Introduction to Algorithmic Design II
Spring 2026
http://marmorstein.org/~robert/Spring2026/cs162.html

Lecture: 2:00 – 2:50pm MWF (Rotunda 356) **Lab:** 2:00 – 3:15pm R (Rotunda 356)

Instructor: Robert Marmorstein **Office:** Rotunda 331
Office Phone: (434) 395-2185 **E-mail:** marmorsteinrm@longwood.edu
Office Hours: 3:00pm – 4:00pm MWF, 10:00am – 11:00am TR, *or by appointment*

To make an appointment, please contact me by Slack or by e-mail with your schedule for the week. Include as much detail as you can about why you need to see me (this saves time). I will reply with some times I am available within the next two days so that we can find a time that works for both of us.

Course Description: A continuation of CMSC 160. Topics include algorithmic design, complexity analysis, abstract data types, encapsulation and basic data structures. Advanced topics using a modern high-level programming language include inheritance, overloading, and use of objects. **4 credits.**

Prerequisite: Grade of C- or better in CMSC 160
Prerequisite/Corequisite: CMSC 161

Textbook : The textbook for this class is "C++20 for Lazy Programmers", 2nd Edition, by Dr. Will Briggs, Apress, ISBN: 978-1484263051.

Student Learning Outcomes: At the end of this course, the successful student will be able to:

– apply object-based principles to creating understandable and maintainable solutions to problems

– identify appropriate implementations for abstract data types such as stacks, queues, lists, sets, trees, and maps

– explain and implement data structures such as linked lists, trees, and hash tables

– compare and contrast standard algorithms using complexity analysis

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

– Explain how C++ implements encapsulation
– Determine whether a class variable is mutable or immutable

Course Structure and Student Expectations: This is a heavily project-driven lecture course that combines theory with pragmatic programming practice. We will spend roughly three hours each week in lecture and discussion sections and one hour a week in class working on laboratory projects.

You should expect to spend at least nine additional hours each week outside of class reading the textbook, preparing for exams, and working on assignments.

University Policies: This course adheres to the university policies found at http://www.longwood.edu/academicaffairs/syllabus-statements/.

Disability Accommodations Policy: If have a disability and require accommodations for this course, I am happy to work with you, but you must (prior to receiving accommodations) do two things:

1. Register with with the Accessibility Resources Office in Brock Hall

2. Schedule a meeting with me early in the semester to discuss a plan for your accommodations.

Should you need temporary accommodations due to an injury or illness, you should reach out to both me and ARO and we can discuss how to meet your needs while you work with ARO to address your long-term needs.

Course Requirements: Your grade will depend largely on completion of the weekly lab sessions. These projects will comprise **50%** of your grade. The remainder of your grade will come from homework assignments and quizzes(**25%**), participation(**5%**), the midterm exam(**10%**), and the final exam(**10%**).

Grading Policy: Numeric grades are translated to letter grades using the following grading scale:

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

There is no grade of D- in this course. Anything below a grade of 64% is failing.

Your final grade in this course is computed using a weighted average of your scores on each assignment. The weights for each category are given in the course requirements section of this syllabus and can be used by applying the following formula:

*Final Grade = 0.50 * Projects + 0.25 * (Homework and Quizzes) + 0.05 * Participation + 0.10 * Midterm + 0.10 * Final*

Each of the category grades (such as Projects) can be computed by summing the points you've earned on each assignment in that category, multiplying by 100 and then dividing by the total number of points possible to obtain a percentage.

Late Work: In general, I do not accept late work or grant extensions on assignments unless you have a serious medical or family emergency which prevents you from completing the assignment on time (however, see "Slip days" below). In exceptional circumstances, I may be persuaded to grant extensions on one or more projects or assignments. In such cases, you do not need a doctor's note, but you must notify me of the circumstances within a reasonable amount of time (typically within twelve hours of the deadline).

All requests for such extensions MUST be submitted by **e-mail** within a reasonable amount of time. This e-mail should outline in detail the reasons your work is late. Granting of extensions is entirely at my discretion – if you have not turned an assignment in on time, you should expect to earn a zero.

Slip Days: You will be allocated a fixed number of slip days at the start of the semester. You may use your slip days to extend the due date of one or more *programming projects*. You can use all of your slip days on one assignment or you may use them over multiple assignments.

Slip days are calculated from the minute the assignment is due until you turn it in. The number of slip days used is rounded *up* to the nearest integer value. That means that if you turn an assignment in 24 hours and 1 minute after the due date, you will use up *two* slip days. The slip day clock runs over weekends and holidays. If a lab is due on Friday and you turn it in on Monday, you will have used three slip days, not one. Slip days cannot be shared, traded, bought, or sold, but can occasionally be earned by participation in relevant campus activities I select.

Slip days may not be used on homework assignments, quizzes, or exams.

Honor Code Policy: I take the honor code very seriously. I encourage you to take advantage of the freedom it offers to collaborate with your class mates. However, there is a point at which collaboration crosses a line and becomes cheating. I treat such infractions harshly under the Longwood honor code and 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.

If you have questions about the honor code policy, PLEASE ask me. It is much better to receive a late penalty on a single assignment than to fail the course and face honor board charges. You may find the scenarios at https://integrity.mit.edu/handbook/writing-code helpful in understanding this policy. While their honor code policy is not identical to mine, it IS similar and the examples may be helpful.

When properly followed the honor code is a tool that sets clear guidelines for what you can and cannot do academically. To that end, here are some principles you should follow in this class:

You **MAY NOT** use Generative AI tools or web sites like "chegg" that advertise complete solutions to homework exercises and projects.

Generative AI systems such as ChatGPT, Gemini, Claude, and Cursor by their inherent nature produce results that use intellectual work scraped off the Internet without proper attribution. As such, use of these systems implicitly commits plagiarism. You **MAY NOT** use these systems in ANY way on work you turn in for this class.

You **MAY** collaborate on homework problems with other students as long as you:

a. *write down (or type) your own answers in your own words* and

b. *give credit to those with whom you have collaborated.*

To give credit, simply write the names of others you have worked with in the margin of your handout.

You **MAY NOT** collaborate with anyone on tests and quizzes.

Tests and quizzes **MUST** be completed entirely on your own. Unless otherwise indicated in class, you should consider all tests and quizzes to be closed-book and closed-notes. You should not discuss them with anyone but me.

You **MUST** give proper credit to sources you use in your work in ANY way (both on your projects and other course assignments).

Plagiarism is a form of cheating that involves taking credit for someone else's work by failing to properly acknowledge the source. This applies not just to papers, but also to class projects, homework, tests, quizzes, and other assignments. Whenever you submit code, text, images, music, sound effects, interviews, video clips, or data for an assignment, you are taking credit for that work. If what you turn in is not 100% original, you should always cite a source.

There is one exception to this rule: you do **NOT** need to cite assistance you obtain directly from me or the textbook. Information I provide to you during office hours or lecture does not need to be cited unless it comes from me only indirectly (for example, if I point you to an article on the web, you do not need to cite me, but **SHOULD** cite the web page).

Students often don't realize that adding a source to a bibliography page or the top or bottom of a document is not a sufficient citation. A correct citation also indicates which parts of the derived work are based in whole or in part on the external source. The easiest way to do this in an academic paper is by adding either a footnote, endnote, or in-text citation at the place where the borrowed material is used.

In code, it is usually best to place a comment citing the source on a line immediately above the borrowed material. *For example:*

/ based on http://codewarrior.com */*

or

/* Jessica helped me with the curly braces here */

is fine.

Students sometimes incorrectly think that they can use a source without citing it if they "tweak it" enough by paraphrasing or rewording it (or in code, changing the names of the variables or reordering some of the instructions). This is incorrect. It is not just the literal words that deserve proper academic credit, but also the ideas behind those words.

The Longwood library has some good materials about how to avoid plagiarism at this link: https://libguides.longwood.edu/c.php?g=1144855&p=8355762.

DO NOT copy large blocks of code from other students or the Internet, even if you cite them properly. You **MAY** assist other students or get assistance with simple problems like syntax errors, but you **MAY NOT** copy large blocks of code, such as entire classes or files, from a web site or from another student. How much code is "too much" depends partly on context, but a good guideline of what "large" means is the three line rule (if you are copying more than three complete programming statements, that is probably too much).

Attendance: I expect you to attend class unless you are sick or engaged in a school-sponsored sport or extracurricular activity. Please do NOT come to class if you are sick.

Instead, contact me within twelve hours of the absence to check whether you've missed any work and make arrangements to make up any missed quizzes. You should also make arrangements to get notes from another student in the class. You should also check the course web site for announcements, new assignments, and other important updates.

I will rely primarily on your honor for enforcement of the attendance policy. However, I will keep a record of your attendance. In accordance with Longwood policy, missing more than 10% of scheduled class time (5 class sessions) to unexcused absences may, at my discretion, result in loss of one letter grade and missing 25% of class or more (14 sessions), whether excused or not may result in an automatic failing grade.

Cell Phones and Laptops: Cell phones, music players, and laptops are to be turned off and put away during class, except as needed for the lab sessions. Violations of this policy will be considered an **unexcused** absence. I will not interrupt class to notify you if you have been counted absent for use of a prohibited device. Feel free to contact me by e-mail at any point in the semester to check on the number of absences you have in my class.

Food and Drink: You may bring non-alcoholic beverages, including soft drinks, to class. However, please do not eat in class (it distracts me and the other students). Violations of this policy will be considered an **unexcused** absence. I will not interrupt class to notify you if you have been counted absent for violation of this policy. Feel free to contact me by e-mail at any point in the semester to check on the number of absences you have in my class.

I occasionally grant exceptions to this rule for students who must otherwise forgo 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 in advance (i.e. before bringing food to class).

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 account and report any missing files, flash drives, or other devices to me promptly.

Major Assignments: In addition to the two exams, there will be homework assignments, quizzes, and several laboratory projects in this course.

Projects: Projects are worth 50% of your grade. There will be six to eight laboratory projects. For tentative due dates, see the course schedule on this syllabus.

Exams: The midterm exam will be taken in class on March 7th and is worth 10% of your final grade. The final exam will be held on Monday, May 6th at 3:00pm and is also worth 10% of your grade.

Homework Assignments and Quizzes: These will largely be drill worksheets intended to help you practice and retain key concepts. The number of points each one is worth will vary with the difficulty of the material and the number of problems. In addition, I will provide you with review materials for the exams that will be considered homework.

I give **unannounced pop quizzes**. These are usually worth 10 to 20 points and count toward the homework grade. Together, homework assignments and quizzes will make up 25% of your final course grade.

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. I recommend you install it on both types of devices. Slack will allow you to easily send me code snippets, ask questions in real time, or set up a Zoom meeting if we need to video chat. You should sign up for a Slack account by visiting

https://longwood-cmsc.slack.com

Use your @live.longwood.edu email address to register and you will be automatically approved for an account.

I will expect you to check the **#cmsc-161** channel every day before class in case I have posted an announcement or asked you to bring something to class.

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. While I am often available in the evening or on weekends, you may need to be patient if I am busy with other students or family obligations.

If you are **asking for help with a project or homework problem**, you should attach your work to a direct message in Slack so that I can see where you are at. You should do this by using the “plus” 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 pictures of your work taken on your phone. These are often blurry and always hard to read. Also, if you attach your code I can run it to see why it is failing, but if you only send me a picture of it, I will have to “guess” why it is wrong. Nevertheless, it is sometimes useful to be able to see a picture of your screen. The best way to do this is to take a screenshot of your system using the “Spectacle” program (usually by pressing the Print Screen “PrtSc” key).

One last suggestion: **don't "ask to ask"**. Asking me whether you can ask a question wastes my time and yours. I am delighted to answer questions about the projects and homework assignments and you should feel free to ask questions at any time (yes, even 3am the night before the project is due – I MIGHT be awake and online – and I'm happy to help you find an answer).

Slack is also a good way to communicate with other members of the class. You will be invited to a public **#cmsc-161** 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.**

I am much slower at replying to e-mail (since I do not get a notification and have to log in to check it). Typically, you can expect a reply to an e-mail within 48 hours, but this may be longer on weekends, and I may not receive your message at all or may not be able to respond to it (my inbox is often over the "quota" allowed by campus I. T. and this often prevents me from using the system effectively).

Tentative Course Schedule:

Week 1: Jan. 14 – 16	Introduction: Data Structures and Abstract Data Types Review: Arrays, Vectors and Maps Strings and C-Strings **Read Syllabus and Chapter 1**
Jan. 15	**Lab 0: C++ and Unix Review (Due Jan. 29)** Review: Preprocessor Directives Review: Command-Line Arguments Review: Files, File Streams, and String Streams **Read Chapter 13**
Jan. 19	**NO CLASS: MLK Jr. Day**
Week 2: Jan. 21 – 23	Structs and Pointers, LValues and RValues, Smart Pointers, and Move Semantics std::swap and std::move Dynamic Memory and Dynamic Arrays

	Read Chapter 14
Jan. 22	**Project Work Day**
Jan. 22	**Last day of Add/Drop (by 5 pm)**
Week 3: Jan. 26 – 30	Classes, Encapsulation, Mutators and Accessors Implementation and Interface Constants and Mutability
	Read Chapter 15
Jan. 29	**Lab 1: Lists (Due: Feb. 12)**
Week 4: Feb. 2 – 6	Array-Based Lists Linked Lists Constructors and Destructors Move Semantics
	Read Chapter 16 and Chapter 22
Feb. 5	**Project Work Day**
Week 5: Feb. 9 – 13	Template Functions and Template Classes Asymptotic Analysis: Big-O / Big-Theta Linear and Binary Search
	Read Chapter 20
Feb. 12	**Lab 2: Templates, Operators and Dynamic Arrays (Due: Feb. 26)**
Week 6: Feb. 16 – 20	Sets and Bags Operators and Exceptions, Operator Overloading Virtual Functions, Libraries and Linking Design Principle: Code Reuse and Modularity **Read Chapter 17**
Feb. 19	**Project Work Day**
Week 7: Feb. 23 – Feb. 27	Quadratic Sorting: Selection Sort, Bubble Sort, Insertion Sort Constant Expressions, Lambda Functions, Casting operators
	Read Chapter 26 (pages 557 – 562 and 578 – 584)

Feb. 26	**Lab 4: Inheritance (Due: Mar. 5)**
Week 8: Mar. 2 – 6	Recursion, Divide and Conquer, Decrease and Conquer Efficient Sorting: Quick Sort, Merge Sort **Read Chapter 18**
Mar. 5	**Project Work Day**
Mar. 9 – 13	**Spring Break: NO CLASS**
Mar. 16	**Exam Review**
Week 9: Mar. 18 – Mar. 20	Variable Scope, Static Members Design Principle: Internal and External Documentation **Read Chapter 26 (pages 566 – 574)**
Mar. 19	**Midterm Exam**
Week 10: Mar. 23 – 27	Stacks, and Queues Inheritance, Abstraction and Polymorphism Design Principle: Code Reuse and Modularity **Read Chapter 19**
Mar. 26	**Lab 5: Stacks, Graphics, and Exceptions (Due: Apr. 2)**
Week 11: Mar. 30 – Apr. 3	Multiple Inheritance Design Principle: Top-Down Design Backtracking, Towers of Hanoi **Read Chapters 21**
Apr. 1	**Deadline to withdraw without an 'F'**
Apr. 2	**Project Work Day**
Week 12: Apr. 6 – 10	Hash Functions, and Hash Tables, Open and Closed Hashing
Apr. 9	**Lab 6: Hashing (Due: Apr. 16)**
Week 13: Apr. 13 – 17	Trees, Tree Traversals, Balanced Binary Search Trees
Apr. 16	**Lab 7: Sorting (Due Apr. 23)**

Week 14: Apr. 20 – 24	Priority Queues / Heaps Heap Sort
Apr. 22	**Research Day: No Classes**
Apr. 23	**Lab 8: Trees (Due Apr. 30)**
Week 15: Apr. 27 – May 1	Iterators, Casting, and Type Deduction Initializer Lists Dynamic Programming Testing and Debugging **Read Chapter 23**
Apr. 30	**Exam Review**
May. 6	**Final Exam (Wednesday, 3:00pm – 5:30pm)**