

CMSC 161: Software Development Tools and Principles (Fall 2025)

<https://marmorstein.org/~robert/Fall2025/161.html>

Lecture (Rotunda 352): 2:00pm – 2:50pm WF

Instructor: Dr. Robert Marmorstein, 395-2185, marmorsteinrm@longwood.edu

Office Hours (Rotunda 331): 12:00pm – 3:00pm TR

Note: To make an appointment, check the schedule outside my office door, then send me a Slack message or email with your availability. Be aware that I typically need 24 hours notice to be able to adjust my schedule.

Course Description: This class is a laboratory-driven class which prepares students for advanced work in computer science by developing facility with the non-programming tools and practices that support software development. Students will learn to collaborate using agile programming practices, use a text editor to create internal and external documentation, navigate the Linux command line environment to run programs and manipulate files. **1 credit.**

Co-requisite: CMSC 160 or CMSC 162.

Textbook: There will be no textbook for this course. Instead, I will expect you to read a series of online resources (such as tutorials) and other handouts that I will provide on the course web site.

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

- Run simple Linux commands and construct command pipelines
- Use software tools and agile programming practices to collaborate with others
- Create and edit program code and documentation using the vim text editor
- Create development and production environments using virtual machines and/or containers

Course Structure and Student Expectations: This is a laboratory-driven class which will meet two times a week for fifty minutes (twenty five total hours). Some class sessions will be lecture-driven and introduce a software tools. These will be followed by laboratory sessions in which you practice and use the tool. You should take careful notes of the lecture and any class discussions or question periods that follow.

You will also need to spend time outside of class completing the projects, reading tutorials and other resources, and preparing for tests and quizzes. The amount of time this requires will vary from week to week, but you should expect this to take around an hour per week on average (about 12.5 to 14 hours total over the course of the semester).

Course Requirements: Your grade will depend on your successful completion of projects, a final exam, participation, homework assignments, and quizzes.

Major Assignments:

Note: Due dates for the major assignments of the class can be found at the end of this syllabus on the tentative course schedule.

Projects: There will five or six projects in this course. Some of them will be individual projects designed to teach you specific skills. Others will be group projects designed to help you learn to collaborate with others. See the tentative schedule below for projected due dates.

Homework Assignments and Quizzes: I give unannounced pop quizzes in class. You should prepare for these quizzes by making sure you have done the assigned reading for each class before coming to lecture.

Homework assignments will be submitted as hard copies to my office (Stevens 109). If I am not in, you may slip them under the door.

Electronic copies of handouts, which you may print if you lose your hard-copy handout, and links to other readings and resources will be made available on the course web site (<http://marmorstein.org/~robert>).

I do not use Canvas for this course. Electronic submission will be through the submit system at <https://marmorstein.org/~robert/submit/>

Final Exam: The final exam will take place on **Wednesday, May 7th** at **3pm**. I will hand out the exam in **Rotunda 352** and it will be due to my office by the end of the exam period. It will be a comprehensive final covering all topics of the course. A study guide will be provided and collected before the exam, but the exam will be closed book and closed notes.

Grading Policy: In general, I do not accept late work and assign it a grade of 0% (even if it is only a few minutes late). However, in some circumstances, such as a medical condition or serious emergency which prevents you from completing the assignment on time, I may be willing to grant an extension. If this arises, you do not need a doctor's note, but you **MUST** contact me by Slack or by e-mail within a reasonable period of time (typically 12-16 hours) to explain the reason for the late work so that I can decide whether it merits an exception to the policy.

Grading 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: 63 and below (note that there is no grade of D- in this course).

Grade Weights: Your grade will depend on your successful completion of laboratory projects (60% of your grade), a final exam (20%), participation (5%), and homework assignments and quizzes (15%).

Attendance Policy: I expect you to attend class unless you are sick or engaged in a school-sponsored extra curricular event (such as a research conference, programming competition, or athletic tournament). I will primarily rely on your honor to enforce the attendance policy, but I do adhere to Longwood's 10% and 25% rules. In accordance with that policy, missing more than 10% of scheduled class time to unexcused absences may, at my option, result in loss of one letter grade. Missing more than 25% of class (whether to excused or unexcused absences) may result in a failing grade.

If you must miss class due to illness or an approved event, you should contact me on Slack or by e-mail **before** the absence (or, if that is not possible, within 12 hours of the absence) to explain why you were absent and to work out a schedule for any missed work (such as pop quizzes).

Disability Accommodations Policy: If you have an approved accommodation for a disability through the Accessibility Resources Office (ARO), you should arrange to meet with me during the first week of class to discuss how we can address your accommodation without drawing attention to you or disrupting the course.

If you have an accommodation to take your exams in an alternative location (such as the distraction reduced environment), you must make arrangements with me at least 24 hours before each exam so that I can provide a copy of the exam to ARO. In general, I do not extend accommodations to students who have not been officially approved by ARO. However, if you have a temporary medical condition (such as a broken leg), you should contact me to make short-term arrangements while you negotiate with ARO.

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. Such images 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 “PrtScr” or “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!).

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

Food and Drink: Please do not eat in class (it distracts me and the other students). You MAY bring non-alcoholic beverages to class. I occasionally make exceptions to this rule for students who would otherwise miss lunch or have a related medical condition. If you feel that you need such an exception, you MUST make arrangements with me before you bring food to class. Violations of this policy will be considered an unexcused absence.

Cell Phones and Laptops: Cell phones and laptops must be turned off and put away during lecture, unless specifically requested by the instructor. Violations of this policy will be considered an unexcused absence and may also incur a grade penalty.

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** collaborate on homework problems with other students as long as:

- 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).

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).

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** use these to generate examples for your own exploration and learning **as long as you are careful to keep that work separate from anything you submit.**

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>.

Additional Policies: Information about additional resources, such as the campus intellectual property statement, accessibility resources, mental health resources, and information on how to report crimes and sexual misconduct can be found at:

<http://www.longwood.edu/academicaffairs/syllabus-statements/>

Tentative Course Schedule:

Week 1: Aug. 27 – 29

Introduction to CMSC 161
Using Slack to Communicate
Linux CLI: Basic Shell Commands
Linux CLI: Reading and Searching Man Pages

Project 0: Introduction to Linux

Due Sept. 3

Sept. 1

Labor Day: NO CLASS

Week 2: Sept. 3 – 5

Connecting to a Linux System (SSH and SFTP)
Linux CLI: Tab Completion and History
Linux CLI: Pipelines and Redirection

Networking: File Transfer (wget and curl)

Week 3: Sept. 10 – 12

Basics of Unix

Vim: Normal Mode, Insert Mode, Replace Mode, Visual Mode
Vim: Navigation and Searching
Vim: Yank, Delete, and Put

Linux CLI: Basic File and Folder Commands
Linux CLI: Managing File Archives with “tar”
Linux CLI: Environment Variables and the Shell Prompt

Project 1: Vim Fundamentals

Due. Sept. 17

Week 4: Sept. 17 – 19

Project Work Day

Vim: Swap Files
Vim: The .vimrc file
Linux CLI: The .bashrc file

Week 5: Sept. 24 – 26

Agile Principles: Internal and External Documentation
Agile Principles: Coding Style and Formatting Tools
Documentation: Markdown, YAML, and HTML
Basics of Web Design

Vim: Code Formatting

Sept. 25

Symposium Day: No classes for seniors in CTZN 410

Week 6: Oct. 1 – 3 Virtual Machines and Docker Containers
Networking: Addresses and Domain Names
Linux CLI: File Permissions and Ownership
Vim: Change, Substitution, and Replace commands

Project 2: Using Containers

Due Oct. 8

Week 7: Oct. 8 Project Work Day

The Compilation Tool Chain
Compiler Flags
Makefiles

Oct. 9 – 10

Fall Break: NO CLASSES

Week 8: Oct. 15 – 17 Linux CLI: Jobs, Processes, and Signals
Vim: Buffers and Windows
Agile Development: Using git to track changes
Vim: Undo/Redo

Week 9: Oct. 22 – Oct. 24 Agile Development: Using git to share files
Resolving Merge Conflicts

Week 10: Oct. 29 – Oct. 31 Project Work Week
Vim: Code Completion

Project 3: Introduction to Git

Due Nov. 5

Week 11: Nov. 5 – 7 Agile Development: Scrum and Kanban
User Stories

Week 12: Nov. 12 – 14 Agile Development: Principles of Software Testing

Project 4: Agile/Kanban

Due Apr. 16

Nov. 19

Research Day: NO CLASSES

Week 13: Nov. 21 Regular Expressions and Globbing
Linux CLI: Using grep and sed

Project 5: Advanced Vim and Unix Commands

Due Dec. 5

Nov. 26 – 28

Thanksgiving Break: NO CLASS

Week 14: Dec. 3 – 5 Final Exam Review and Project Work Week
Dec. 12th Final Exam (Friday, 8:00am – 10:30am)