CMSC 461: Senior Capstone in Computer Science (3 credits)
Spring 2026

https://marmorstein.org/~robert/Spring2026/cs461.html

Instructor: Robert Marmorstein, 395-2185, marmorsteinrm@longwood.edu
Lecture: 10:00am – 10:50am, Stevens 118
Office Hours: 3:00pm – 4:00pm MF, 10:00am – 10:50am TR or 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 or Slack and list some possible times we could meet. Typically, I will need at least twenty-four hours of notice, so plan accordingly.

Course Description:
A capstone course designed to consolidate experiences from a variety of other courses by working in groups on one or more large projects. Principles of software engineering will be covered, including traditional and object-oriented software design, software life-cycle models, software analysis, and management implications. The implementation of developing software using teams will be stressed along with various software tools. Re-usability, portability, and interoperability will be discussed. A segment on assessment will be included. 3 credits.

Prerequisites: CMSC 208, CMSC 262.

Textbook:
This course has one required textbook: "Beginning Software Engineering", Rod Stephens, Wrox Publishing, First Edition (2015), ISBN: 978-1-118-96914-4

Course Student Learning Outcomes:
This course has three central themes: **review** of topics from throughout the major, **software engineering** (the ability to collaborate to develop large software projects), and **career preparation**.

At the end of the course, a successful student will be able to:

* Describe the elements of at least two Agile Programming paradigms

* Collaborate with peers to create and deploy a functioning, tested, and properly documented software product

* Show mastery of computer science principles, such as data structures and algorithms, grammars, state machines, and networking concepts

* Demonstrate mastery of algorithms, data structures, programming language theory, computer organization, and other computer science topics on a standardized test.

Course Structure and Student Expectations:
You should expect to spend at least three hours each week in class for lecture, discussion, and project work sessions and another six or seven hours outside of class working on the semester project with your group, preparing for exams, or completing review exercises.

Part of this "out of class" time should be reserved for a **daily** stand-up meeting with your project team. This means that for about ten to twelve weeks of the course, you will need to set aside about twenty minutes every week day (even days the class does not meet) to meet with your group.

Grading Policy:
Late work will not be accepted unless you have a medical condition or family emergency which prevents you from completing the assignment on time. **There will be no slip days in this course.**

In the event of an emergency, you should contact me within twenty-four hours of the due date with details of why you were unable to complete the assignment. At my discretion, I may then extend the due date, give an equivalent alternative assignment, or resolve the situation as events dictate.

Grading Scale:
Letter grades will be assigned using the following scale. Note that there is no D- grade for this class. Any grade below a 64% is failing.

	91 – 100:% A	90%: A-
89%: B+	81 – 88%: B	80%: B-
79%: C+	71 – 78%: C	70%: C-
69%: D+	65 – 68%: D	0-64%: F

Note: A failing grade on the semester project automatically results in an F for the class.

Communications Policy:
The best way to reach me is to use **Slack**. You should install Slack on both your phone (if you have one) AND your laptop. 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 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-461 channel in which you can discuss the projects and other course topics with other students in the class. Feel free to ask for and give help on this channel, but please stick to general answers rather than posting code. I will expect you to check the #cmsc-461 channel for messages **at least once a day**.

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. Instead, attach the .cpp or .h file directly to the Slack message. You will probably need to do this from a browser running inside your Linux virtual machine (or other Linux system).

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 permission to ask a question wastes my time and yours.

Attendance:
I expect you to attend class unless you are sick or engaged in a school sponsored sports event or extra-curricular activity. I will rely on your honor to enforce the attendance policy. In accordance with Longwood policy, missing more than 10% of scheduled class time to unexcused absences may result in loss of one letter grade. Missing more than 25% of class (whether excused or unexcused absences) may, at my discretion, result in a failing grade.

Food and Drink:
I prefer that you 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 a meal or who have medical needs. 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 I have specifically requested, usually by e-mail, that you bring them to class (e.g. for a lab day). Violations of this policy will be considered an unexcused absence.

Major Assignments:
A significant part of this class comes from a single group project which you will complete outside of class. The assignment is designed to simulate, as closely as possible in an academic setting, a real world software project.

You should be prepared to spend a minimum of six hours a week working on the semester project. While you will have some in-class time to work on projects, you will need to budget time outside of class to complete your assigned tasks, communicate with your team, and evaluate your work.

In particular, you should plan to set aside **twenty minutes every week day** (not just days we have class) for a daily scrum meeting or “stand-up”. These meetings should be strictly time-boxed to take no more than 15-20 minutes and will give everyone on your team a chance to report progress on their tasks, identify blockers, and set priorities.

The second major component of this course is preparation for a standardized test designed to evaluate the effectiveness of our program at teaching you computer science fundamentals. For this purpose, we will use the “ETS Major Field Test” (MFT) as the final exam for our course. This is a nationally standardized computer science test. We will also complete a review test designed by the faculty of this department. This test will serve as review for the MFT and will count toward your homework grade.

Throughout the semester, I will assign reading and homework assignments selected to prepare you for these tests. These review assignments will comprise the majority of your homework grade and are largely taken from practice tests for the MFT and for other nationally standardized assessments. Each student in the class will be responsible for preparing a summary of the topics covered by these questions and leading a review discussion explaining the solutions.

The in-class portion of the class will consist partly of lecture on software engineering topics and partly on student-led discussions of review topics. Over the course of the semester, I will assign one or two questions from each practice test to each student. You will be responsible for leading a roughly ten to fifteen minute discussion of the review questions. You should be prepared to spend at least an hour preparing your presentation.

The third major theme of the course is career preparation. We will spend some time in class talking about the job market and I will expect you to update your resume outside of class to submit for my review.

Campus Policies:
This course adheres to the campus policies listed at
http://www.longwood.edu/academicaffairs/syllabus-statements/.

If you have course accommodations through the disability services office, please contact me outside of class to discuss how we will implement those in this course.

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.

Course Requirements:
A significant part of your grade (60%) will be earned by completing the semester project. **Failure to complete this project successfully will result in a failing grade even if your numeric grade is otherwise high enough to pass the course.**

The remainder of your grade will come from homework assignments and quizzes(25%), participation (5%), and performance on the MFT, which will serve as our Final Exam(10%).

Unlike many of my other courses, the homework grade may include both written work and short programming exercises. These, as well as the review packets and pop quizzes, will count toward the homework grade.

Honor Code:
I take the honor code seriously in my classes. Students suspected of an honor code violation will be charged with honor offenses. Any student convicted of an honor offense will receive an F in the course in addition to any penalties imposed by the honor council.

All work in this class should be considered pledged. Tests and quizzes must be completed entirely on your own and will be taken closed-book and closed-notes. You *may* discuss homework problems and laboratory projects with other students subject to the following restrictions:

1. Your submitted work must consist of *your own answers in your own words* which you have typed or written yourself. You may discuss assignments verbally with other students, but do not share code or answers electronically.

2. You must acknowledge any help you receive from anyone outside your group, including any discussion of the homework problems, by leaving a short note in the margin of the assignment, or in the case of a project, placing appropriate comments in the code. Such acknowledgments should indicate which section or sections of your work you have discussed.

3. Do not copy large blocks of code or directly copy answers from other students, the Internet, or other resources. You can discuss the general approach to an assignment and you can help other students find syntax errors in their code, but any block of code longer than two or three lines should be entirely your own work.

Tentative Course Schedule:

Jan. 14 – 16: **Introduction to Software Engineering, Resumes and Interviews**
Applications for Product Owner and Scrum Master by Jan. 16th

The Software Life Cycle, The Mythical Man-Month
Writing a good resume, How NOT to interview for a tech job

Read Chapters 1 and 2
Read Sample Resumes

Jan. 19: Martin Luther King Jr. Day (NO CLASS)

Jan. 21 – 23: Agile Software Development
Product Owners: Submit Job ads for Developer and Support Positions by Jan. 21st

Read "Using GitLab to facilitate Scrum"
(https://docs.gitlab.com/ee/tutorials/scrum_events/)

Read Chapter 14

Scrum, Kanban, Extreme Programming, LEAN Development

Jan. 22: Last day of Add/Drop (by 5pm)

Jan. 26 – 30: Requirements and Specifications

Applications for Developer and Support Roles	**Due Jan. 26th**

Read Chapters 3 and 4
Read "Software Requirements Specification Format"
https://www.geeksforgeeks.org/software-requirement-specification-srs-format/

Requirements Gathering, Use Cases, User Stories

Feb. 2 – 6: Project Management and Metrics

Sprint One begins	**Feb. 2**

Read Chapter 10

PERT Charts and Gantt Charts, Software Metrics, Project Metrics

Feb. 9 – 13: Project Work Week

MFT Review Test One	**Feb. 9**
Sprint One Ends	**Feb. 13**

Feb. 16 – 20: Software Design

Sprint One product demonstration	**Feb. 16**
Sprint Two Begins	**Feb. 16**
Sprint One Retrospective	**Feb. 20**

Read Chapters 5 and 6

High-Level Software Design, Low-Level Software Design

Feb. 23 – 27: Software Development and Testing

MFT Review Test Two	**Feb. 22**
Sprint Two Ends	**Feb. 27**

Read Chapters 7 and 8

Development Tools, Top-Down vs Bottom-Up Design
Data Structures and Algorithms Review

Smoke testing, Unit testing, Integration Testing, Usability Testing, Performance Testing, Testing Tools, Continuous Integration/Continuous Delivery

Mar. 2 – 6: Software Deployment

Sprint Two product demonstration	**Mar. 2**
Sprint Three Begins	**Mar. 2**
Sprint Two Retrospective	**Mar. 4**

Read Chapter 9

Mar. 9 – 13: SPRING BREAK (NO CLASS)

Mar. 16 – 20: Project Work Week

MFT Review Test Three	**Mar. 16**
Sprint Three Ends	**Mar. 20**

Mar. 23 – 27: Software Maintenance

Sprint Three product demonstration	**Mar. 23**
Sprint Four Begins	**Mar. 23**
Sprint Three Retrospective	**Mar. 25**

Read Chapter 11

Mar. 30 – Apr. 3: Process Models (Waterfall, Iterative, Agile)

Sprint Four Ends	**Apr. 3**

Read Chapters 12 and 13

Apr. 1: Deadline to Withdraw without an F (by 5pm)

Apr. 6 – 10: Project Work Week

Sprint Four product demonstration	**Apr. 6**
Sprint Five Begins	**Apr. 6**
Sprint Four Retrospective	**Apr. 8**

Apr. 13 – 17: Project Work Week

Sprint Five Ends	**Apr. 17**

Apr. 20 – 24: Project Final Presentations

Presentation and Demonstrations	**Apr. 24**

April 22 **NO CLASS: Research Day**

Apr. 27 – May 1: **Computer Science Program Assessment Test**

MFT Review Test Four	**Apr. 27**

Final Exam: May 4 **MFT Exam (Monday, 11:30am – 2:00pm)**
Location TBA