

CMSC 355: Software Engineering: Specification and Design

Fall, 2021

Syllabus

Catalog Listing: CMSC 355 - Software Engineering: Specification and Design

Course Level: Undergraduate

Prerequisites: Students must have completed CMSC 256 or EGRE 246 with a grade of C or better

Instructor: Dr. David C. Shepherd

Office: The Internet

E-mail: shepherdd@vcu.edu

Classroom: [YouTube Live](#)

Office Hours: M 2:30pm – 4:30pm

1.0 – Overview (Catalog Course Description):

Semester course; 3 lecture hours. 3 credits. Prerequisite: CMSC 256 or EGRE 246, either with a minimum grade of C. Provides an overview of the software engineering process and software life-cycle models. Gives a detailed study of the analysis, specification and design phases. Students will work in teams to gain experience in software development methodology, developing specification and design documents and developing a prototype.

2.0 – Course Structure:

- Lecture hours/week – 3
- Lab hours/week – 0

3.0 – Course Goals

Upon successful completion of this course, the student will be able to:

1. Understand the software lifecycle and different software development methodologies
2. Express requirements and design of a software system
3. Work as a team to develop software products using agile software development methodologies
4. Understand software quality and be able to effectively test software
5. Perform software maintenance and use appropriate tools
6. Use effective software architectures and design patterns

4.0 – ABET Criteria Addressed:

- a. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
- b. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
- c. An ability to function effectively on teams to accomplish a common goal

5.0 – Major Topics Covered:

- Software development lifecycle
- Framework of the software process
- Requirement engineering
- Software architecture and design
- Software modeling via UML
- Software verification and validation

- Agile software development
- Design patterns
- Software maintenance

6.0 – Textbook(s):

No required textbook. Extensive notes and online reading materials will be provided.

7.0 – Class Schedule:

- Lecture: M,W,F 1:00pm – 1:50pm, YouTube Live

8.0– Evaluation:

General instructions:

There are few individual assignments in this course, but the majority of assignments are team projects. Significant amount of time outside class meetings will likely be required for the successful completion of the projects, including ample time spent communicating and planning with the team. Students will be graded on their individual contribution to the team project, using a combination of peer and individual ratings in the project status reports. Several mechanisms will be used to detect students that fail to contribute significantly to the team project; those students will not get full credit for their team's project(s).

Grading:

<i>Category</i>	<i>Percentage Weight</i>
Midterm Exams (x2)	30%

Project	40%
In-Class Exercises	30%

Grades will be posted on Canvas in a timely manner, and include appropriate feedback and weights.

Attendance policy:

Attendance is expected from all students in the course. To encourage attendance, the instructor will use periodic, short, in-class exercises, which will be graded as either acceptable or non-acceptable. While the grade for 1-2 of these questions will be dropped from final calculation, a student will not be allowed to complete them after class, even in the case of a justified absence.

Grading scheme:

A: $\geq 90\%$

B: $\geq 80\%$ and $< 90\%$

C: $\geq 70\%$ and $< 80\%$

D: $\geq 60\%$ and $< 70\%$

F: $< 60\%$

Please consult external resources for VCU policies regarding academic honesty, students with disabilities, student conduct in the classroom, withdrawal from classes, and others.

Class Schedule

Week	Date	Topic	Activity
0	Aug 25	Course logistics; What is software engineering?	
	Aug 27	Programming day	
1	Aug 30	Software development processes; Intro to agile development	
	Sept 1	Introduction to Android	
	Sept 3	Programming day	
2	Sept 8	Group assignments and logistics	
	Sept 10	Introduction to Android (2)	
		Programming day	
3	Sept 13	Version Control Systems	
	Sept 15	Behavior-Driven Design and User Stories	
	Sept 17	Programming day	
4	Sept 20	User Stories and Backlogs - Examples	

	Sept 22	Storyboarding: Agile Project Tracking	
	Sept 24	Programming day	
5	Sept 27	Git Review	
	Sept 29	Class review; Intro to Testing	
	Oct 1	Programming day	
6	Oct 4	TDD	
	Oct 6	Test Doubles ; UI Testing in Android	
	Oct 8	Programming day	
7	Oct 11	Software Design: Software Architecture	
	Oct 13	Software Modeling: UML ; UML practice day	
	Oct 15	Programming day	
8	Oct 18	Service-Oriented Architecture	
	Oct 20	SOA, Part 2	
	Oct 22	Programming day	

9	Oct 25	Midterm Exam #1	
	Oct 27	Software as a Service	
	Oct 29	Programming day	
10	Nov 1	SOLID Design Principles ; Post-Midterm Discussion	
	Nov 3	SOLID Design Principles	
	Nov 5	Programming day	
11	Nov 8	Observer Design Pattern	
	Nov 10	Template Method Design Pattern	
	Nov 12	Programming day	
12	Nov 15	Big Ideas in Software Dev	
	Nov 17	Big Ideas in Software Development ; Performance Tuning	
	Nov 19	Programming day	
13	Nov 22	Code Reviews	
	Nov 24	Debugging; Secure Coding Principles	

		Programming day	
14	Nov 29	Debugging	
	Dec 1	Debugging II	
	Dec 3	Static Analysis	
		Dynamic analysis	