

CS321 Syllabus: Software Requirements/Design Modeling

CS 321 gives an introduction to principles and techniques used in software engineering:

- An understanding of all phases of the software engineering lifecycle (requirements, design, implementation, testing, deployment, maintenance).
- An understanding of several software lifecycle models including both prescriptive and agile models and knowledge of tradeoffs among these models.
- An ability to create and use UML models to develop and document software analysis and design artifacts.
- An understanding of fundamental project management and leadership issues for software teams.
- An ability to apply software engineering methods to address the requirements and design phases of a large software project.

Course Outcomes

CS 321 will have a software engineering project that requires student to participate in working teams where students organize, manage, and practice a software engineering project. This will be a design project with some prototype implementation. See the [project page](#) for more details.

CS 321 includes Writing Intensive (WI) activities that, together with those of CS 306, meet the GMU WI Requirements in the BS CS Program (<http://wac.gmu.edu>). This means you will write 1750 graded words (or about 7 standard pages). You will get feedback on this writing, and be able to resubmit revisions based on the feedback. For this course, part of the writing will include an individual essay on your experiences working with your team.

Prerequisite C or better in CS211

**Professor
Contact
Information** Dr. Kinga Dobolyi
Office: Engineering (Rm. 4440)
Email: kdobolyi@gmu.edu
Phone: 703-993-4198

<http://www.cs.gmu.edu/~kdobolyi/cs321/>

Office Hours: MTWR (1:45 PM — 2:45 PM) or by appointment. Please email the professor at least 48 hours in advance

when seeking an appointment.

**TA Contact
Information**

Upsorn Praphamontripong
Email: uprapham@masonlive.gmu.edu
Office: Engineering (Rm. 5321)
Office Hours: Wednesdays (1:30PM — 3:30PM)

Textbook

Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, *Fundamentals of Software Engineering*, 2nd Edition. (optional: it is NOT required that you purchase a copy of this book for the course. I have a copy in my office if you want to refer to it. Instead, we will be reading several scholarly articles throughout the semester.)

**Course
Tools**

- **Discussion board:**
[Piazza](#)
- **Subversion (SVN) tools:**
[Tortoise SVN](#) - SVN client for Windows only
[SCPlugin](#) - SVN client for Mac OSX (but NOT Snow Leopard)
[svn](#) - command line (Windows version in link) (Mac version built in to OSX)
- **Project Hosting tool:**
[XP dev](#)
- **UML Tools:**
[Netbeans](#) with UML plugin
[Violet UML](#) - Open source UML tool - Simple, looks good, but sometimes hard to get exactly what you want
[ArgoUML](#) - Open Source UML tool - Previous classes did not like this tool, but it has more features than Violet
[IBM Rational Software Architect](#) - Very mature UML and architecture tool
- **Project Scheduling Tools:**
[OpenProj](#) - Open source version of MS Project
[Microsoft Project](#) - Industry standard project scheduling tool

Email

I will use email to send information about the course, ask questions, etc... You must check your GMU email account. If you prefer to use another account, that is okay, just forward your GMU email to your preferred account.

- Project assignments (40%)
- Essay (10%)
- Midterm (25%)
- Final Exam (25%)

Final course grades are calculated as follows:

Grading

A+ ($\geq 98.0\%$) A ($\geq 92.0\%$) A- ($\geq 90.0\%$)
 B+ ($\geq 88.0\%$) B ($\geq 82.0\%$) B- ($\geq 80.0\%$)
 C+ ($\geq 78.0\%$) C ($\geq 72.0\%$) C- ($\geq 70.0\%$)
 D ($\geq 60.0\%$)
 F ($< 60.0\%$)

Contesting of grades on any/all submissions must be requested within one week of the item's return. No grade changes will be considered subsequent to that deadline, or after the final exam meeting.

Group members must participate equally in project assignments; the professor reserves the right to assign a zero to any student for any deliverable retroactively, if it turns out that nothing of academic merit was contributed to the assignment by any individual. It is up to students to prove their participation on deliverables, and the professor will examine ONLY svn logs in the event of any disputes of participation. Make sure you are using svn, and using it through your login.

Students are expected to attend and actively engage in lectures. Some lectures will involve completing reading assignments at home and being prepared to discuss them (or take a quiz) in class. Lectures are also used to solve in-class examples and work on the semester project.

Class

Participation

Groupwork will make up a large part of the students' grades. Students are expected to participate equally and fully in group assignments. Each deliverable will require students to complete the deliverable all, or in part, individually at first. The instructors will examine these individual assignments and adjust points for groupwork if students are found to not contribute adequately to draft deliverables.

Late Policy

All assigned must be received by the deadline (submitted on Blackboard by 11:55pm on the date due). No late assignments will be accepted. All project assignments must be submitted online, through your team's repository, by 11:55pm on the day

they are due.

Exams

Exams are closed book/notes unless specified otherwise by instructor. Appropriate documentation (as determined by instructor) and requisite permissions are required for make-up exam requests. Picture IDs are required to take all exams. If you know in advance that you are unable to make an exam for a valid and unavoidable reason (such as a scheduled surgery, etc), you must notify the professor at least one week before the scheduled exam date to make arrangements for a make-up, and bring documentation with you when you take the make-up. If you miss an exam due to a university-accepted excused absence (such as an illness or car accident the day and time of the exam), you must notify your professor within 24 hours of your absence to make arrangements for a makeup, and bring approved documentation with you when you take the make-up exam. Failure to follow either of these policies will result in a zero. You must take the final exam for the section that you are registered for.

Honor Code

All students are expected to abide by the [GMU Honor Code](#). This policy is rigorously enforced. All class-related assignments are considered individual efforts unless explicitly expressed otherwise (in writing). Review the university honor code and present any questions regarding the policies to instructor. Cheating on any assignment will be prosecuted and result in a notification of the Honor Committee as outlined in the GMU Honor Code.

Learning Disabilities

Students with a learning disability or other condition (documented with GMU Office of Disability Services) that may impact academic performance should speak with professor ASAP to discuss accommodations.

Tentative Class schedule

Week #	Date (start of week)	Lecture Topics	Readings and homework (to be completed by dates posted - everything is due at 11:55pm unless otherwise noted)
		<ul style="list-style-type: none"> Class overview, project 	

1	1/21	<p>overview</p> <ul style="list-style-type: none"> Discuss this Boehm paper Software engineering overview Version control demo - take notes! <p>Spend 10 minutes in class trying to find a team</p>	<ul style="list-style-type: none"> Read the Quiz Game semester project description (see link above) <hr/> <ul style="list-style-type: none"> Complete the Team Formation deliverable (submitted to XP-Dev) found on the project page
3 Updated due to snow day and prof absence	2/3	<ul style="list-style-type: none"> Use Cases use case template Requirements 	<ul style="list-style-type: none"> Team: Team formation deliverable (due 2/10) Team: Use Cases Diagram deliverable (due 2/17) Team: Use Cases Descriptions deliverable (due 2/24) <hr/> <ul style="list-style-type: none"> Team: Requirements deliverable (due 3/3) Homework 1: Requirements Homework 2: Use Cases
3	2/10	<ul style="list-style-type: none"> Static analysis modeling (class diagrams) BookstoreExample4.jpg 	<hr/> <ul style="list-style-type: none"> Team: Class diagram deliverable (due 3/17) Homework 3: Class Diagrams
4	2/17	<ul style="list-style-type: none"> Finish class diagrams Dynamic analysis modeling 	<hr/> <ul style="list-style-type: none"> Team: Dynamic analysis deliverables (due 3/24) Homework 4: State Diagrams
		<ul style="list-style-type: none"> Design 	

5	2/24	<ul style="list-style-type: none"> • Prescriptive software process models • Agile software process models 	<hr/>
6	3/3	<ul style="list-style-type: none"> • Project Management and Scheduling • schedule example • Start Testing techniques 	<hr/>
7	3/17	Midterm Review MIDTERM EXAM THURSDAY (confirmed, voted on by class)	<ul style="list-style-type: none"> • Team: implementation (due 3/31) <hr/>
8	3/24	Finish testing techniques <ul style="list-style-type: none"> • Unit testing 	<hr/> <ul style="list-style-type: none"> • Individual: Read this paper on Findbugs by next week • Team: Unit tests (due 4/14)
9	3/31	Go over the midterm <ul style="list-style-type: none"> • Static analysis example research paper: Findbugs 	<ul style="list-style-type: none"> • Individual: Run the Findbugs tool on the source code provided by the professor for the semester project. How many bugs did it flag? For each bug, describe the warning, and then decide if it should be fixed, or not, and why. Write a 250 minimum word summary of your experiences reading the Findbugs paper and using the tool (part of this summary should address the questions just posed). Submit your assignment on BB (due 4/21, 15 points) <hr/> <ul style="list-style-type: none"> • Individual: essay outline (due 4/11; see project page) Example Essay

- Individual: Read this [paper on readability](#) by next week.

11

4/7

- [Metrics](#)
- Metrics example research paper: [readability](#)

- Team: System test cases (due 4/16)
-

12

4/14

- [Risk Management](#)
- Technical writing instruction
- [Refactoring and Code Smells](#)

-
- Individual: essay draft (due 4/22 - in class)
 - Team: presentations start next week!

13

4/21

- [Sample final](#)
- **Start in-class presentations**

- Individual: Run your unit tests on your project code using one of the code-coverage tools mentioned in class. Record the percentage of code covered in both cases. Were your unit tests effective? Why or why not? What coverage criteria did you originally use when writing your unit tests? How does that relate to code coverage? Write a 500 word summary of your experiences using unit tests and coverage criteria on this exercise. Attach a screenshot of running your unit tests on the codebase. Submit your assignment on BB (due 4/29)
-
- Individual: Complete the code smells quiz on BB. The quiz will become visible on May 1, and you will have only one chance to submit the assignment before the final exam.

14

4/28

- **In-class presentations**

Final Exam: see the standard time on the [George Mason Academic Calendar](#). The exam will be given in the regular classroom.