

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: Tuesday 12:00pm—2:00pm, Thursday 12:00pm—1:00pm, or by appointment. Please email the professor at least 48 hours in advance when seeking an appointment.

Comment [GG1]: Syllabus contains WI statement

Comment [GG2]: Students are required to write 1750 words (this course is ½ of the CS WI course requirement).

Comment [GG3]: Syllabus describes that the instructor will provide feedback on writing.

Comment [GG4]: Students are required to write at least 1 assignment (this course is ½ of the CS WI course requirement).

**TA Contact
Information**

Please see Piazza for GTA information

Textbook

None required.

Course Tools

- **Discussion board:**
[Piazza](#)
- **Subversion (SVN) tools (free):**
[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)
- **UML Tools:**
[Netbeans](#) with UML plugin
[Violet UML \(free\)](#) - Open source UML tool - Simple, looks good, but sometimes hard to get exactly what you want
[ArgoUML \(free\)](#) - 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 \(free\)](#) - 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.

- Quizzes (20%)
- In-class assignments (5%)
- Project assignments (40%)
- [Essay \(10%\)](#)
- Final Exam (25%)

Comment [GG5]: Syllabus describes grading criteria for all major writing assignments.

Grading

Final course grades are calculated as follows:

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.

This course uses a flipped classroom approach.

I have provided links to content on Youtube as a supplement to materials. The linked content is not owned by me, nor was it created by me, therefore I don't have any rights or control over what exists in the videos - view them at your own risk. Having said that, these are the popular and useful videos I found on Youtube for various lecture topics. Feel free to look for ones on your own, or let me know if you found a better one than anything I linked. If you do not want to view the videos, you do not have to; the slides contain all the information you're responsible for for quizzes, and the in-class exercises will allow you to practice the material and ask questions.

Class Participation

Students are expected to attend and actively engage in lectures. Most lectures will involve completing reading assignments at home and being prepared to take a quiz in class. Lectures are also used to solve in-class examples and work on the semester project. Lectures will not generally be used to introduce or review material, as that should be done in reading (or watching videos) at home.

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. This class will have a large discussion-based component. I expect discussions to be civil, polite, and reasonable. Any student that acts in a rude, offensive, or disrespectful manner towards another student or the professor will automatically have their final grade in the course dropped half a letter grade without warning.

Late Policy

All assignments must be received by the deadline (submitted on Blackboard by 11:55pm on the date due if not otherwise noted). 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.

Quizzes

We will have a short, five minute quiz at the beginning of almost every lecture, on reading material assigned for that day. The quizzes will be open-book, individual assignments, and will take the place of a midterm exam.

Final Exam The final exam is closed book/notes.

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.

Other Useful Resources [Counseling and Psychological Services](#) (CAPS): (703) 993-2380
[Writing Center](#): A114 Robinson Hall; (703) 993-1200
[Ask a Librarian](#) through the University Libraries

Tentative Class schedule

Date (start of week)	Lecture Topics and Notes	Readings and homework (to be completed by dates posted - everything is due at 11:55pm unless otherwise noted)
9/1	<ul style="list-style-type: none"> Class overview, project overview Spend 10 minutes in class trying to find a team. [10:30am section] Software engineering overview 	<ul style="list-style-type: none"> Read the Boehm paper for next class
9/3	<ul style="list-style-type: none"> Quiz on Boehm paper Boehm paper on software engineering 	<ul style="list-style-type: none"> Team Due (9/7): Complete the Team Formation deliverable (submitted through Piazza) found on the project page Individual Due

		(9/8): Prepare Use Cases topics (watch video, review slides)
9/8	<ul style="list-style-type: none"> • Quiz on use cases • In-class use cases exercises (found at end of slides) • Use Case Introduction video • Use Cases • use case template 	<ul style="list-style-type: none"> • Individual: Due (9/9): Use case homework - finish exercises from class, if needed • Individual: Due (9/10): Prepare Requirements topics (review slides)
9/10	<ul style="list-style-type: none"> • Quiz on requirements • In-class requirements exercises (found at end of slides) • Requirements 	<ul style="list-style-type: none"> • Individual: Due (9/11): Requirements homework - finish exercises from class, if needed • Individual: Due (9/15): Prepare class diagram topics (watch video, review slides)
9/15	<ul style="list-style-type: none"> • Quiz on class diagrams • In-class class diagrams exercises (found at end of slides) • Class diagram introduction video • Static analysis modeling (class diagrams) 	<ul style="list-style-type: none"> • Individual: Due (9/16): Class diagram homework - finish exercises from class, if needed
		<ul style="list-style-type: none"> • Team: Due (9/18): complete use cases diagram and descriptions

9/17	<ul style="list-style-type: none"> • Version control demo - take notes! • Work on team assignments for use case diagrams • Work on team assignments for use case descriptions 	<p>deliverables, if needed</p> <ul style="list-style-type: none"> • Watch appropriate version control video as needed: Mac/command line (start at 2:16, go to 4:45 video), or Windows GUI (video)
9/22	<ul style="list-style-type: none"> • Work on team assignments for requirements 	<ul style="list-style-type: none"> • Team: Due (9/23): complete Requirements deliverable, if needed • Individual: Due (9/24): Prepare dynamic analysis topics (watch videos, review slides and reading)
9/24	<ul style="list-style-type: none"> • Quiz on dynamic analysis diagrams • In-class dynamic analysis diagrams exercises (found at end of slides) • Swimlane diagram video (first 5 minutes only) . • State diagram video (starting at minute 5:30) . • Sequence diagram tutorial chapter (through figure 10) . (sorry, Youtube videos were horrible) • Dynamic analysis modeling 	<ul style="list-style-type: none"> • Individual: Due (9/25): Dynamic analysis diagram homework - finish exercises from class, if needed
9/29	<ul style="list-style-type: none"> • Work on team assignments for class diagrams 	<ul style="list-style-type: none"> • Team: Due (9/30): finish Class diagram deliverable, if needed • Individual: Due (10/1): Prepare design topics (watch videos, review slides and

		reading)
10/1	<ul style="list-style-type: none"> • Quiz on design • In-class design exercises (found at end of slides) • Factory Design Pattern video (stop at 2:30) • Singleton Design Pattern video (stop at 3:40) • Visitor Design Pattern video (stop at 9:00) • Single Responsibility Principle video (stop at 4:30) • Interface Segregation Principle video (stop at 4:30) • Design 	
10/6	<ul style="list-style-type: none"> • Work on team assignments for dynamic analysis diagrams 	
10/8	<ul style="list-style-type: none"> • Continue to work on team assignments for dynamic analysis diagrams 	<ul style="list-style-type: none"> • Team: Due (10/9): finish Dynamic analysis deliverables, if needed • Individual: Due (10/15): Prepare management topics (watch videos, review slides and reading - only starred links)
10/15	<ul style="list-style-type: none"> • Quiz on management (just starred items) • Software Scheduling video* • Project Management and Scheduling* • schedule example • Prescriptive software process models • Agile software process models 	<ul style="list-style-type: none"> • Individual Due (10/20): Imagine you have the method <code>string triangleType(int side1, int side2, int side3)</code>. Write an exhaustive set of test cases, as you would have done for CS211, for this method.

10/20	<ul style="list-style-type: none"> • In-class testing exercises • Start Testing techniques 	<ul style="list-style-type: none"> • Individual: Due (10/22): Study testing topics from today
10/22	<ul style="list-style-type: none"> • Quiz on testing • Finish Testing techniques • Unit testing demo (if necessary) 	
10/27	<ul style="list-style-type: none"> • In-class pair programming 	
10/29	<ul style="list-style-type: none"> • In-class pair programming, system and unit testing 	<ul style="list-style-type: none"> • Team: Due (10/30): finish Implementation, if needed • Team: Due (10/30): finish System and Unit Test cases, if needed • Individual: Due (11/3): Read Findbugs paper for next class
		<ul style="list-style-type: none"> • Individual: Due (11/5): Watch metrics video (slides will be discussed in class) • Individual: Due (11/7): Run the Findbugs tool on the source code provided by the professor for the semester project. How many bugs did it flag? For

11/3	<ul style="list-style-type: none"> • Quiz on Findbugs • Static analysis example research paper: Findbugs 	<p>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 (8 points)</p>
11/5	<ul style="list-style-type: none"> • Quiz on metrics video • Software metrics video • Metrics 	<ul style="list-style-type: none"> • Individual: Due (11/10): Read readability paper for next class.
11/10	<ul style="list-style-type: none"> • Quiz on readability paper • Metrics example research paper: readability. 	<ul style="list-style-type: none"> • Individual: (Due 11/12): Prepare risk topics (watch videos, review slides and reading)
11/12	<ul style="list-style-type: none"> • Quiz on risks • In-class risk exercises • Risk Management • Risk management paper 	<ul style="list-style-type: none"> • Individual Due (11/13): Risk management homework - finish exercises from class, if needed • Individual: Due (11/17): essay outline in class (see project page)

11/17	<ul style="list-style-type: none"> • Technical writing instruction; Example of high-quality writing • Review essay outline in class - meet with professor to discuss writing (5 pts) 	
11/19	<ul style="list-style-type: none"> • Begin essay writing in class - meet with professor to discuss writing (5 pts) 	
11/24	<ul style="list-style-type: none"> • Work on essays - meet with professor to discuss writing if you've not met yet 	<ul style="list-style-type: none"> • Individual: (Due 12/1): Prepare refactoring topics (watch videos, review slides and reading) • Individual: (Due 12/3 - in class): essay draft
12/1	<ul style="list-style-type: none"> • Quiz on refactoring • Refactoring Introduction video • 15 refactoring demos in Eclipse • Refactoring and Code Smells 	
12/3	<ul style="list-style-type: none"> • Guest speaker James Dressel from Agilex 	
12/8	<ul style="list-style-type: none"> • Begin final review • Sample final 1 • Sample final 2 • Sample final 3 	
12/10	<ul style="list-style-type: none"> • Finish final review 	

Comment [GG6]: Syllabus describes that the instructor will provide feedback on writing.

Comment [GG7]: Syllabus indicates multiple class sessions that address different writing strategies.

Comment [GG8]: Syllabus indicates required draft deadline with sufficient time for thoughtful faculty feedback and for student revision.

Final Exam: see the standard time on the [George Mason Academic Calendar](#). The exam will be given in the regular classroom. In addition to the exam, your team should bring a laptop to demo your

semester project.