Syllabus for SE 14-322: Engineering Software Intensive Systems
Department of Software Engineering
Achi Racov School of Engineering
Kinneret College on the Sea of Galilee

Instructor: Michael J. May

Semester 2 of 5786

1 Course Details

The course meets **10:00am–2:00pm** on Thursdays. The course has **4** hours of lecture. The course will take place in Room 817.

2 Overview

This course leads students on the transition from the world of programming (writing computer programs) to the world of *software engineering* where we develop *software intensive systems (SIS)*. We will focus on the development life cycle - requirements, analysis, design, implementation, integration, testing - while maintaining a systems approach. Students will also perform exercises in model based software development using the Unified Modeling Language (UML).

The course will be taught as a "seminar class" in which lecture and practice are interwoven. A typical session will consist of 2-3 hours of lecture and 1-2 hours of student active practice of the techniques presented. Some sessions will be entirely dedicated to practice.

2.1 Course Goals

By the end of the course, students will be able to:

1. Define life cycle options (Waterfall, V, Agile) for the development of software intensive systems.
2. Build a requirements table categorized by requirement type (functional, non-functional) and class (*e.g.* hardware, implementation)
3. Create Use Case Diagrams and Use Case details for software intensive systems
4. Design a set of logical software components based on a user story and a set of use case details
5. Allocate logical components to physical components as part of a software intensive system deployment
6. Create and document the physical, logical, and composite architectures of software intensive systems.
7. Create a plan for the development of object oriented programs based on classes and object interaction.
8. Create, understand, and update UML diagrams for all of the above tasks.

3 Lecture Schedule

The course lectures are structured in the following way.

#	Topic	Subjects	Readings
1	Introduction	What is software engineering?	[Sch11]1–2
		Software's special issues	[Bro86]
2	Developing SIS	Software from a Systems Perspective	[RJB05]2, [Sch11]2
		Development life cycles, & patterns, modelling	[SSHK15]1
	Requirement elicitation in SIS	Gathering & categorizing requirements	[Sch11]11[PM20]10.5,
3	Requirement elicitation and management in SIS	Managing requirements, Building a requirements table,	[PM20]11.2 [RJB05]5
	System process definition	Use Cases	[Sch11]11
4	System process definition	Use Case Identification, Use Case Diagrams and Specification	[Sch11]11
5	System process definition	Use Case Specifications	[Sch11]11
		Activity Diagrams	[RJB05]7, [Sch11]17.8
		Activity Model and Business Logic	
6	System Physical Architecture	Physical architecture	[RJB05]9, [Sch11]17.11
7		Design review 1	
8	Functional analysis and	Decomposition to components	[RJB05]7–8, [Sch11]15
	Process Definition		
9	Functional analysis and Process Definition	Sequence models	[SSHK15]6–7
	Software Architecture	Component models	[RJB05]9
10	Overall System Architecture	Composite model	[RJB05]9
	Object Oriented Software Design	PDOM, Class model	[PM20]21, [Sch11]17
11		Design review 2	
12	Object Oriented Software Design	Class Model, OOP	[PM20]2, 10
13	Testing	Static testing patterns	[PM20]18.4–6, [Sch11]6, 15.20–22
	Design improvement principles	Design patterns, SOLID	[PM20]22.4, [Sch11]8

4 Semester project

During the course of the semester, you will develop a project in stages. There are two submission milestones for the project which will be followed by design reviews. Both design reviews will take place during the lecture slot.

The project will involve the design and partial development of a software system. There will be two inter-

mediate design reviews in class and a final submission. The grade will be a combination of grades from the in-person design reviews and the final project submission.

The project can be done in teams of 4 or 5 students. More details about the project will be given out during the course of the semester.

5 Attendance

Students are responsible for all material presented in class, recitation, and laboratory sessions, all assigned readings, and all material provided for additional reading out of class. Students who miss a lecture or targil can look at the course syllabus and web page to see which material was missed.

Attendance of lectures and targil sessions is expected and required for this course. As per College policy, a student who misses 20% or more of the lectures of targil sessions may be expelled from the course. Students who miss lectures do so at their own risk and expense and will be expected to make up missed material on their own. Students who know they will be missing two or more lectures due to circumstances beyond their control should inform the instructor as soon as possible beforehand.

5.1 Decorum

Students who attend lecture are expected to give their full attention to the material. Reading newspapers, talking on cellular phones, text messaging, or other distracting behavior will not be tolerated.

Students must arrive to lectures **on time, within the first 5 minutes of class.** After ten minutes into class, the door will be locked and no student will be allowed entry. The door will be opened at the next break in the lecture (approximately every 50 minutes). Students who need to leave during lecture for some urgent matter must leave quietly and may return at the next break.

As per college policy, the instructor reserves the right to expel from the classroom any student who is disturbing the lecture or others.

6 Submissions

6.1 How to Submit Work

To ensure timely submission of projects and work, students must submit all work via Moodle. Materials sent via email or via any other method risk being ignored or ungraded without consideration of their merits.

When submitting project work via Moodle, every student on the team **must participate** in the submission. If a student's name appears on a submission, but the student doesn't make any substantive contribution to the work or does not upload the submission on Moodle, the student **will not** receive a grade for the assignment.

6.2 Late Submission Policy

Students are expected to be on time with their project submissions and assignments. Each assignment must be turned in by the date it is due. Students who miss design review submission dates will receive a 0 for the design review. An member of a student team who misses the design review presentation will not receive a grade for the design review stage.

Students who are called up to Miluim duty will have their assignment deadlines extended in accordance with college policy.

7 Cheating

Cheating of any sort will not be tolerated. Student collaboration is encouraged, but within limits as set forth in the college's rules on academic integrity. Any students caught cheating will be immediately referred to the department head and the Dean and may receive a failing grade for the course.

Cheating includes:

- Copying information, content, or verbatim text from other students, internet sites, books (other than the ones listed in the bibliography), other unaffiliated individuals to answer questions, solve problems, or aid in programming projects.
- Copying or submitting source code, documentation, or other programming aids **without attribution** from other students, **web sites**, online repositories, text books, open source programs, or other unaffiliated individuals.
- Project teams which submit work which is identical or substantially identical to work submitted by other project teams, whether current or from previous years.
- Other forms of academic misconduct as described on the site: `https://catalog.upenn.edu/pennbook/code-of-academic-integrity/` or as reasonably assessed by the instructor, program head, or dean.

If you have any questions about what constitutes cheating in the above rules, contact the instructor as early as possible.

8 Exams

There is no exam.

9 Grading

Final grades will be calculated by combining grades from programming project stages. The grades are weighted as follows:

15%	First design review
15%	Second design review
70%	Project final submission

The instructor will not address questions about specific individual grades during the lecture or targil sessions. Students may contact the instructor *in person* during office hours or after the lecture/targil sessions at the instructor's convenience.

10 Books

The following books are used for the class:

- Pressman and Maxim. *Software Engineering: A Practitioner's Approach.* [PM20]
- Rumbaugh, Jacobson, and Booch. *Unified Modeling Language Reference Manual.* [RJB05]
- Schach. *Object Oriented and Classical Software Engineering.* [Sch11]
- Seidl, Scholz, Huemer, and Kappel. *UML @ Classroom: An Introduction to Object-Oriented Modeling.*[SSHK15]
- Gomaa, Hassan. *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures.*[Gom11]

The library has copies of the books listed, but students are encouraged to purchase the books as needed.

A bibliography of the books and articles used in the course of the semester is shown below.

11 Contact Information

Instructor:	Michael J. May
Web page:	`https://mjmay-kinneret.github.io/`

References

[Bro86] Fred P. Brooks. No silver bullet — essence and accident in software engineering. In *Proceedings of the IFIP Tenth World Computing Conference*, pages 1069–1076. IFIP, 1986.

[Gom11] Hassan Gomaa. *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures*. Cambridge University Press, New York, N.Y, 1 edition, 2011.

[PM20] Roger Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education, 9th edition, 2020.

[RJB05] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual*. Addison-Wesley, 2nd edition, 2005.

[Sch11] Steven Schach. *Object Oriented and Classical Software Engineering*. McGraw-Hill Education, 8th edition, 2011.

[SSHK15] Martina Seidl, Marion Scholz, Christian Huemer, and Gerti Kappel. *UML @ Classroom: An Introduction to Object-Oriented Modeling*. Undergraduate Topics in Computer Science. Springer, 2015.