

LARGE-SCALE SOFTWARE DEVELOPMENT – SYLLABUS

CSc 260, Winter 2023

Aaron G. Cass
Union College

ADMINISTRATIVE DETAILS

COURSE WEB SITE

<http://cs.union.edu/csc260>

CLASS MEETINGS

The course will meet from 3:05pm to 4:45pm on Mondays and Wednesdays in Wold 225.

INSTRUCTOR INFORMATION

name: Aaron G. Cass
e-mail: cassa@union.edu
web site: www.cs.union.edu/~cassa
office: Steinmetz Hall Room 220
office phone: 388-8051
home phone: 382-9671 (before 10pm please)

OFFICE HOURS

If you need help, or just want to chat, please:

- Come by during my scheduled office hours. See <http://cs.union.edu/~cassa/schedule.html> for up-to-date info on my office hours.
- Just stop by, especially if you think it will be quick.
- Schedule a meeting, especially if you think it will not be quick. To schedule a meeting with me, go to <https://cs.union.edu/~cassa/schedule.html> and follow the easy instructions. If you and a classmate have similar questions, feel free to schedule a meeting together with me.

COURSE DESCRIPTION

From the college catalog:

Strategies for the systematic design, implementation, and testing of large software systems. Design notations, tools, and techniques. Design patterns and implementation idioms. Implementation, debugging, and testing. Includes team and individual software development projects. Prerequisites: Math 197 and C- or better in CSc 150.

GOALS

In this course, you will learn and use tools and techniques that are commonly used for the development of large software systems. This will involve design modeling, implementation of those design models, and verification of the resulting implementations. Specifically, your goals in this course should be:

- Become familiar with terminology used in software development.

- Develop skill in communicating designs using accepted design notations, including class diagrams, interaction diagrams, state transition diagrams, and activity diagrams.
- Develop skill in using development technologies, particularly for configuration management, design diagramming, testing, and building.
- Develop skill in making design decisions to create flexible software.
- Develop skill in using design patterns to design and communicate about design.
- Develop skill in using tools that support large-scale software development.
- Further develop skill in understanding the designs and implementations of others.
- Become comfortable discussing design.
- Further develop your implementation, debugging, and testing skills; in particular, to hone your prototyping and refactoring skills.
- Understand ethical implications of software development work, especially testing.

PREREQUISITES

In **CSc 151: Data Structures**, you developed the programming and design skills of abstraction and encapsulation. In this course, we will build on those skills and see their application to larger software projects.

TEXTBOOKS

REQUIRED

Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995

This is the *classic* book on design patterns. It briefly introduces the concept of design patterns and follows with a collection of forty very commonly-used software design patterns.

RECOMMENDED

As most of the development will culminate in Java programs, you may wish to have ready access to a good reference book on Java. I recommend the Core Java series of the O'Reilly Nutshell series.

You might also consider purchasing:

Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Third. Addison-Wesley Professional, 2004

This book gives you all you need to know to effectively use UML diagrams to document your design choices.

EVALUATION

DEVELOPMENT PROJECTS(63%)

There will be three (3) software development projects, each with multiple deliverables. The first one will be individually-developed while the later ones will be developed in teams of approx. 3 students. Development is not just programming, but also design and testing, so you should expect to give presentations on all aspects of your projects.

FINAL EXAM(35%)

There will be a final examination. You will be asked to answer questions about given designs, solve design problems given requirements, and generally to demonstrate your grasp of the material from the class.

SEMINARS (2%)

The CS department has a regular seminar that explores the diverse and rich field of computer science. Our seminar speakers discuss a variety of topics, including computer science research, applying for CS jobs or graduate school, and career and technical skill development. The seminars are frequently presented by visitors from academia and industry as well as alumni. The seminar is typically Thursday during common lunch (12:50-1:45pm) with a lunch provided at 12:30pm and is held in Olin 107. Seminar dates and topics are announced in class and on the CS department mailing lists.

As part of taking an upper-level CSC course (numbered 250 or higher), you are required to attend **at least one** CS seminar this term.

You are strongly encouraged to attend as many CS seminars as you are able, as they will give you context necessary to propose and present a strong senior thesis, and prepare for life involving computing after Union – even if your major is not computer science!

CAVEATS

- Students taking more than one upper-level CSC course are required only to attend and summarize a single seminar each term. This exception does not apply to students in CSC 497, 498, or 499 who are independently required to attend and summarize all CS seminars.
- If you belong to a major whose own mandatory seminars conflict with the times of all computer science seminars, you may instead attend those and submit a summary discussing a seminar you attended and noting the scheduling conflict.

GROUND RULES

ATTENDANCE

I expect you to come to all class meetings.

READING

I expect you to come to class have already read the week's assigned reading. See the attached schedule for the assigned readings.

LATE WORK AND MAKE-UPS

It is important that project deliverables are completed and submitted on time, so that you can proceed to the next (sub-)project and so that I can grade them in a reasonable time. Therefore, project deliverables submitted after their due dates and times will earn a zero.

Essay assignments submitted up to 24 hours late will be penalized 30%. Essays submitted more than 24 hours late will earn a zero.

If you will be unable to take an exam on the scheduled date, you must let me know **ahead of time**, so we can schedule a different exam for you (perhaps before others take theirs). If you are unavoidably detained because of illness or family crisis, please let me know as soon as possible.

ACCOMMODATIONS

I encourage students with disabilities to discuss with me, during the first two weeks of the course, appropriate accommodations that might help facilitate your learning. You will need appropriate documentation

from the Office of the Dean of Students. All discussions will remain confidential.

ACADEMIC INTEGRITY

GENERAL STATEMENT

Union College recognizes the need to create an environment of mutual trust as part of its educational mission. Responsible participation in an academic community requires respect for and acknowledgement of the thoughts and work of others, whether expressed in the present or in some distant time and place.

Matriculation at the College is taken to signify implicit agreement with the Academic Honor Code, available at honorcode.union.edu. It is each student's responsibility to ensure that submitted work is his or her own and does not involve any form of academic misconduct. Students are expected to ask their course instructors for clarification regarding, but not limited to, collaboration, citations, and plagiarism. Ignorance is not an excuse for breaching academic integrity.

Students are also required to affix and sign the full Honor Code Affirmation, or the following shortened version, on each item of coursework submitted for grading:

I affirm that I have carried out my academic endeavors with full academic honesty.

SPECIFIC GUIDANCE FOR THIS COURSE

In this course, you will learn by doing. If you do not do things for yourself, you will not learn from them. Therefore, I expect you to do your own work, and only turn in that which is yours. When you have questions, feel free to talk to me, the help desk students, or even other students in the class. However, do not leave these discussions with just an answer – you need to *understand* how to arrive at the answer.

For **projects**:

- **DO** your own work.
- **DO** struggle on your own before seeking help.
- **DO** seek help (after first giving a serious, honest attempt) from Help Desk, your professor, your fellow classmates.
- **DO** help your classmates by having conversations about general strategies.
- **DON'T** help your classmates by telling them what code to write.
- **DON'T** look at someone else's code, except when you are trying to help *them*.
- **DON'T** give your code to anyone else (on paper, electronically, or in any other way).
- **DON'T** type or write code for anyone else.

- **DO** ask your professor if you are unsure what's permissible or not.
- **DO** put the Honor Code Affirmation in a comment at the top of each program file you submit.

For **exams**, of course, you should work completely alone.

TENTATIVE SCHEDULE

Week	Topic/Reading	Assignments
1	Introduction – How are real projects different from what you’ve done before? – The Graph ADT	W: P1 Out
2	Testing – Invariants – Unit Testing – Quick intro to build and versioning tools	M: P1(a) In W: P1(b) In
3	Refactoring – Prototyping – Design Patterns – Event-Driven Programming <i>Reading: Chap. 1, Chap. 2, Abstract Factory, Factory Method</i>	M: P1(c) In
4	Callbacks, Anonymous classes, lambda expressions – MVC – Design diagramming <i>Reading: Observer, Command</i>	M: P1 In, P2 Out
5	More design patterns – P2 Design presentations <i>Reading: Iterator, Strategy, Visitor</i>	W: P2(a) In
6	Team version control and configuration management – Group work	
7	Group work – P2 Presentations	W: P2 In, P3 Out
8	Test doubles – Group work	
9	Group work – P3 Design Presentations	W: P3(a) in
10	Group work – P3 Presentations	F: P3 In
Finals	Final Exam.	