CS 440, Introduction to Software Engineering, Course Policies
Fall 2018

Course Objectives: At the end of this course, students will be able to:

- Participate fully as a valuable team member on large complex software projects.
- Explain the overall management concepts and issues of large software projects, including the full software life cycle of large projects, from project inception through retirement.
- Specify the requirements for a software product.
- Design a software system, using a variety of software design strategies.
- Implement a software design developed by others, using agile methods & modern CASE tools.
- Verify and validate software systems.

Instructor: John Bell, jbell@uic.edu, 921 SEO, 413-9054
http://www.cs.uic.edu/~jbell
Office Hours: May be M 12:30 – 2:00 and TTh 10:30 – 12:00
- See schedule on web
- Open Door policy during other times.

Teaching Assistant: Ashkan Rezaei, arezae4@uic.edu
and Tanima Chatterjee, tchatt2@uic.edu
- See web site for details

Prerequisites: CS 342

If you are an undergraduate student who does not have the necessary pre-requisites, DROP THE CLASS NOW. Otherwise you will be automatically dropped later, when it will be too late to sign up for anything else instead. Graduate students are not allowed to take CS 440 after or concurrently with CS 442.

This course requires a significant amount of computer programming, as part of a team project, most likely in C++ or Java. All students are expected to know at least one of these languages, regardless of background or graduate status.

Credits: 3 for undergraduate students; 4 for graduate students.

Course Web Page: http://www.cs.uic.edu/~i440


Other Good Books:
1. Robertson & Robertson, “Mastering the Requirements Process” (Available on UIC Library's Safari system.)
2. Gamma Erich, et. al., "Design Patterns – Elements of Reusable Object-Oriented Software" (Also available on Safari)
3. Fowler, Martin and Kendall Scott, "UML Distilled, 3rd Edition".
4. Pezze & Young, “Software Testing and Analysis”
5. Si Alhir, Sinan, "UML in a Nutshell", O'Reilly.
Planned Schedule:

A detailed schedule of activities and assignment deadlines will appear on the course web site, to be updated as the course progresses. In addition, the following exam dates are expected.

MIDTERM EXAM I: Friday 19 October, during class time, Location TBD.
MIDTERM EXAM II: Friday 9 November, during class time, Location TBD.
FINAL EXAM: Time and Location TBD.

Term Project:

This course is primarily concerned with large complex software engineering projects which require multiple programmers and significant time to complete. In order to practice these concepts, a large group project will be conducted throughout the term. This project will serve as the focal point for all material covered in class. Periodic assignments will be due throughout the term, including intermediary results and progress reports.

This semester you will actually complete one-half each of two projects, in parallel: You will plan a project to be implemented at some time in the future, and you will implement a working prototype based on a project previously planned and designed by former students.

Peer evaluations and weekly meeting minutes and activity logs will ensure equal participation and commitment to group activities. Group project grades will initially be assigned with the assumption that all members of a group participate equally. The instructor reserves the right to make adjustments if participation proves to be drastically uneven.

Exam Policy:

- Any exam conflict needs to be brought to the instructor’s attention for resolution before the regularly scheduled exam. Requests for make-up exams after the regularly scheduled exam will not normally be granted.
- Exams will be closed-book and closed-notes.
- All exams will be cumulative, with emphasis on material which has not been covered on previous exams.
- All material covered in class or in assigned reading or which should have been learned in the course of completing homework is fair game on exams. No more specific information will be provided as to exam content.
- Anyone who fails to stop working on their exam when time is called will receive a minimum of a five point late penalty.

Special Considerations

- Assignments submitted after the due date and time, but within 24 hours, will be assessed a 20% penalty. No assignments will be accepted more than 24 hours after the due date and time.

- All appeals for grading errors, no matter how justified, must be submitted within two weeks after the graded assignments are returned. No appeals for regrades will be heard after that time.

- Any student that does not participate fully in their project activities may have their project scores reduced at the end of the term, which will result in a grade reduction.
Grading Policy:

Numerical scores will be based upon the following contributions:

**Individual Work**
- (2) Midterm Exams: 10 points each
- Final Exam: 20 points
- Initial project brainstorming report: 5 points
- Presentation Evaluations: 5 points total
- iClicker points: 5 points total

55 points

**Coding Project**
- (2) Scenarios: 5 points each
- (2) Demonstrations: 10 points each
- Testing and Inspection Report: 10 points
- Weekly iceScrum log: 5 points total

45 points

**Development Project**
- (3) Major Reports: 10 points each
- Description Presentation: 10 points
- Weekly Minutes: 5 points total

45 points

**Final Presentation / Demo** (Both projects combined): 10 points

Other assignments and/or quizzes: as needed

Total: Normalized to 100 point scale

- At any time during the semester your Percentage of Possible Points, PPP, can be calculated as:

\[ PPP = \frac{\text{Points accumulated to date}}{\text{Possible points to date}} \times 100\% \]

- Mapping of PPP score to letter grades will not necessarily follow the traditional 90/80/70/60 grade breaks, however some effort will be made to keep it close. Detailed feedback including an estimate of projected letter grades will be made after each of the midterm exams is graded and returned.

Academic Integrity:

Students are encouraged to study together and to help each other learn. When one student teaches another, both benefit from the experience.

However, it is a strict violation of class and university policy for any student to hand in any work that is not 100% their own creation. Therefore:

- All work on all exams and all individual homeworks must be individually performed by the student whose name appears on the paper.
- No student may give any other student any portion of their work, either written down, electronically, or through any other means.
• Students may discuss homework problems, including background concepts and general solution strategies, but they are forbidden from discussing or sharing specific solutions. In particular, it is forbidden for any student to show any other student any portion of their computer programs or homework solutions for any reason, including debugging assistance. This means you must hand in your own homework. You are not allowed to see anyone else's work, or show your work to anyone. Failure to protect the privacy of your work may be a violation. ♫

♫ Note: The above two points obviously do not apply to group members working together on group assignments. However they do forbid group members from working together on individual homework assignments.

• In the case of extreme discrepancy between homework performance and exam performance (e.g. very high homework scores and very low exam scores), the instructor shall determine which scores more accurately reflect the students' true work.

• First violations will be immediately assigned a NEGATIVE score, for all parties involved in the transgression, and may also be penalized with a grade reduction and/or failure. Second or more serious violations will be reported directly to the Vice Chancellor for Student Affairs, and may result in a failing grade, probation, suspension, or expulsion from the university, as well as being documented on the permanent records of all students involved.