CSCI-4210 Fall/2004

Introduction to Software Engineering

This course is an introduction to software engineering. Life cycles methods, and tool support. The course will concentrate on the aspects of design, implementation and testing issues of Object Oriented systems in particular and software development in general. Tools covered in detail are UML support tools, Testing tools and Software configuration and version control tools. The course will include a project to be developed as a group of 3-4 students.

Course objectives:

• Analyze factors involved in software development. Discussion of the nature of software development process, management and risks.
• Introduce Use-cases to manage requirements.
• Introduce software life-cycle, life-cycle models, project and life-cycle matching.
• Analyze design issues that arise in OO software development and their solutions.
• Introduce refactoring, its place in software development, and introduce a set of common refactorings.
• Introduce software configuration and tools to support it.
• Introduce the concept of frameworks and its place in software design and development.
• Work in a small team, cooperating on these aspects of software development, and exchanging ideas in a constructive and organized fashion;
• Appreciate engineering issues in the development of software, such as the importance of addressing the user’s concerns, working with limited resources, maintainability, dependability, and division of labor.

Outcomes

• Develop a set of requirements for a software project.
• Analyze a set of requirements to develop an initial use-case set.
• Design programs: Construct a design consisting of a collection of modules, using module dependency diagrams to express and evaluate couplings between modules, and object models to relate concrete to abstract state; Exploit well-known design patterns (such as Iterator, Observer, Factory, Visitor, etc.); devise appropriate module specifications and express them informally in terms of pre- and post-conditions; understand the appropriate roles of subtyping and inheritance, and use them effectively.
• Implement programs: Write object-oriented code to satisfy the specification of a module; Express rep invariants, understand their impact on efficiency and ease of implementation, and implement them as runtime assertions; evaluate the correctness of a module by careful manual review using the specification, rep invariant and abstraction function; develop unit and system-level test suites, and evaluate their effectiveness using simple notions of specification and code coverage. Refactor code to improve its design.
• Perform software development in a team, including problem analysis, design, implementation, and testing.
• Acquire skills in the use of software tools such as version control systems, testing systems, ide’s and others.

**Syllabus:**

First 3 weeks or so:
- Introduction to software engineering.
- Introduction to risks in software engineering.

----- *Below are topics for the rest of the semester:*
- Introduction to Analysis of specifications
- Analysis and Design in OO systems.
- Cohesion and coupling of classes.
- Fundamental software design principles.
- Refactoring software and introduction to patterns.
- Software configuration. Version Control. RCS.
- Frameworks as reusable software products.

----- *Topics below only if time allows:*
- Introduction to software architecture.
- Fundamental patterns found in software architectures.

**Textbook:**
These two books should be found in the UNO bookstore.

Other books to look into via the library:
Software engineering. Sommerville.
I will provide a list of reference books for software design in class.

**Office Hours:**

My office is MATH 331. Office hours are from 10:15-11:00 am. M,W, F. Please try to adhere to office hours, or make an appointment if you cannot.

**Email/Home Page:**

You can contact me at jaime@cs.uno.edu. My phone: 280-7362. The course web page is http://www.cs.uno.edu/jaime/Courses/4210. I expect you to read these pages on a daily basis. Usually I place homework, readings and other announcements there which might not be announced in lectures.
From time to time I will communicate through email with you using your department email account; you are responsible to read your email in a timely manner. Also, if you send me email
1. best choice, do it using your department account.
2. write as subject: CSCI-4210 student.
Do not expect me to open nor to answer student email without the specified subject heading.

**Exams:**

Exam schedule is tentative. We will probably have two in-class tests. They are currently scheduled for October 6th, and November 19th. The final exam is scheduled for December 6th, 7:30-9:30 am. No makeup tests or exam will be given.

**Homework:**

We will have many small design/implementation homework; and one large project to work on throughout the whole semester. The project design and implementation must be presented in class as part of its grade. Although the project is done as a group of up to 4 members, all members are responsible to present, and defend the entire development of the project. The project members will grade each of the other members in the team for the work done in the project; this grading is done in private in my office. I will use these team member grades to assign an individual’s grade in the project. Based on project presentation, work submitted, and grades from team members, two given students may have substantially different grades in the project. All homework must be submitted at class time on due date. They can be left in my mailbox in MATH 311, but at your own risk. Late submissions will incur a penalty of 20% per day and I will generally grade late homework more rigorously and grade it at my convenience; that means that you may not get it back until the end of the semester. Homework will not be accepted more than 3 days late. As you should know by now, unless explicitly stated otherwise, homework assignments are individual projects. Use of material other than class textbooks must be referenced.

**Grading:**

In-class tests 20%, final exam 30%, Homework and project: 50%. Letter grades will be assigned as [90 – 100] = A, [80 – 89] = B, etc.

Finally, I must call your attention to the University’s policies regarding academic dishonesty. (See pages 44-47 of the Student handbook.) Academic dishonesty includes cheating, plagiarism, and collusion. In particular, it includes “the unauthorized collaboration with another person in preparing an academic exercise” and “submitting as one’s own any academic exercise prepared totally or in part for/by another.” In the event of academic dishonesty, the student will be assigned a grade of 0 on the exam or exercise, the student will be informed in writing of the action taken, and a copy of this letter will be sent to the Assistant Dean for Special Student Services.

---

**November 8, Final date to drop or resign from University**