

# CS 3398: Software Engineering Spring 2014

Sections 251 and 252

**Instructor:** Dr. Jill Seaman  
Nueces 221  
js236@txstate.edu

**Course Webpage:** <http://www.cs.txstate.edu/~js236/cs3398>

**Office Hours:** M: 12:00pm – 1:00pm  
T: 2:30pm – 3:30pm  
W: 11:00am – 12:00pm  
R: 2:30pm – 4:30pm  
and by appt.

**Meeting Time/Place:** MW 3:30PM-4:50PM DERR 240 (Section 251)  
TH 12:30PM-1:50AM DERR 240 (Section 252)

**Text:** Software Engineering by Ian Sommerville, Addison-Wesley, 9th edition.  
ISBN: 0137035152

**Required readings:** Chapters 1-9

Also recommended:

Essentials Of Software Engineering, Frank Tsui, Orlando Karam, Barbara Bernal  
Jones & Bartlett, 3rd edition, ISBN: 1449691994

## Prerequisites:

- C or higher in CS 3358: Data Structures
- C or higher in CS 2315: Computer Ethics or EE 2400

Note: If you do not meet the prerequisites, you must drop the course.

**Course Description:** The study of software design, implementation, and validation techniques through team projects. Structured analysis, programming style, and project documentation are emphasized in large software projects.

**This is a Writing Intensive course.**

**Course Objectives:** see last page.

<b>Grading:</b>	Assignments:	10%
	Team Project Presentation:	10%
	Team Project Paper (SRS):	15%
	Term Paper:	15%
	Midterm:	25% Mar 5 (W) & Mar 6 (Th)
	Final Exam:	25% See next page

**Final Exam Dates:** Section 251 (MW): Wed, May 7, 2:00PM to 4:30PM  
Section 252 (TH): Tues, May 6, 11:00AM to 1:30PM

**Attendance:** I record attendance every day and I expect you to be in class every day. However, it is not part of the calculation of your final grade.

**Assignments:** This portion of your grade is based on written homework assignments, in-class group exercises and discussions.

**Team Project+Presentation:** I will create teams of 5 people. The main job of the team is to produce a Software Requirements Specification (SRS) document given a problem description that will be provided to you. Each team will give a presentation of their document to the class, and then rewrite and submit the document based on feedback from the presentation.

**Term Paper:** This is an individual assignment. The paper will be not more than 10 pages, and will be a research paper based on three scholarly sources.

**Makeup Policy:** Assignments cannot be made up. Exams may be made up in exceptional circumstances, with approval from the instructor.

**TRACS:** We will use the TRACS website for the following:

- Grades (Gradebook2 tool)
- Term paper submission (Assignments tool)
- Restricted documents (Resources tool)

Everything else will be on the class webpage (including lecture presentations)

**Notifications from the instructor:** Notifications related to this class will be sent to your Texas State e-mail account. Be sure to check it regularly.

**Withdrawals/drops:** You must follow the withdrawal and drop policy set up by the University and the College of Science. You are responsible for making sure that the drop process is complete.

<http://www.registrar.txstate.edu/registration/drop-a-class.html>

**Last day to drop: March 20, 2014.**

**Academic Honesty:** You are expected to adhere to both the University's Academic Honor Code as described [here](#), as well as the Computer Science Department Honor Code, described here: [2013 0426 HonestyPolicy CSPPS.doc](#).

**Classroom Behavior:** Do not disrupt other students during class. Please do not arrive late or leave early (without notifying the instructor).

**Accommodations for students with disability:**

Any student with special needs requiring special accommodations should inform me during the first two weeks of classes. The student should also contact the office of disability services at the LBJ student center.

## Course Objectives:

At the end of the course, the students should be able to:

1. List and describe essential attributes of good quality software.
2. Describe the details of several software development process models.
  - Waterfall, incremental development, spiral, RUP, agile: XP + scrum (with emphasis on software quality and project timeliness).
3. List and describe characteristics of good quality software requirements.
4. Gather user requirements and create software specifications (SRS) from them.
5. List and describe characteristics of good quality software designs.
6. List and describe different architecture design patterns:
  - Pipe+filter, MVC, Client-server, Layered, Repository, Event-driven
7. List and describe different software design techniques:
  - Functional decomposition, Object-oriented Design (UML class diagrams)
8. Produce high quality designs for (simple) software systems.
9. Evaluate the quality of a software system design.
10. List and describe different implementation techniques:
  - Programming style, Comments, Debugging, Refactoring
11. List and describe different verification and validation techniques:
  - Testing, Inspections/reviews, Formal methods
12. List and describe different methods for deriving test-cases:
  - Equivalence class partitioning, Boundary value analysis, Path analysis, etc.
13. Apply verification and validation techniques to produce high quality software.
14. List and describe different types of software maintenance.
15. Describe the software evolution process.
16. Create high quality software systems using a software development process and good software engineering techniques.
  
17. Write technical and non-technical materials for a wide range of audiences.
  - Convey central ideas effectively and critically evaluate sources.
  - Use good structure and logical organization.
  - Use proper mechanics and grammar.
  - Write at a level that is appropriate for the audience.
  
18. Communicate verbally while using appropriate multimedia tools.
  - Present information that is organized logically (and in an interesting manner).
  - Demonstrate proper subject knowledge when speaking.
  - Use good eye contact, a clear voice, and proper delivery when speaking.
  - Use audio-visual materials properly when speaking
  
19. Work successfully in a team environment.
  - Contribute research information for the team.
  - Demonstrate understanding of the assigned team roles.
  - Share the work of the team.
  - Demonstrate good listening skills.