

## **CPSC 4900 Software Engineering**

**SEMSTER:** Spring 2014  
**COURSE:** Section 0, CRN 25613  
**TITLE:** Software Engineering  
**CREDIT:** 3  
**ROOM:** EMCS 220  
**TIME:** TR 10:50 am-12:05 pm  
**INSTRUCTOR:** Li Yang, EMCS 314A  
**PHONE:** 425-4392  
**OFFICE HOURS:** T: 8:50-10:50am, 1:30-4:00pm, W: 8:50-10:50am, R:  
8:50-10:50am, 1:30-3:00pm  
**EMAIL:** [Li-Yang@utc.edu](mailto:Li-Yang@utc.edu)

**ADA STATEMENT:** Attention, if you are a student with a disability (e.g. physical, learning, psychiatric, vision, hearing, etc.) and think that you might need special assistance or a special accommodation in this class or any other class, call the Disability Resource Center (DRC) at 425-4006 or come by the office, 102 Frist Hall  
<http://www.utc.edu/Administration/DisabilityResourceCenter/>.

If you find that personal problems, career indecision, study and time management difficulties, etc. are adversely affecting your successful progress at UTC, please contact the Counseling and Career Planning Center at 425-4438 or  
<http://www.utc.edu/Administration/CounselingAndCareerPlanning/>.

### **COURSE DESCRIPTION:**

Study of techniques used in the definition, specification, design, implementation and testing of large software systems. The course will include team efforts to identify and define the requirements of a large software product. The development of this product will continue in CPSC 4910. Prerequisite: CPSC 3200 with minimum grade of C or department head approval. Differential course fee will be assessed.

### **COURSE OBJECTIVES:**

To introduce students to creating software using a systematic approach. The course will provide an opportunity to develop a software product in a team setting. Specific objectives include:

- 1) Ability to produce a requirements specification document for a complex software product
  - a) Students will be able to identify and describe the major activities that constitute requirements engineering.

- b) Students will be able to describe and perform several different tasks related to requirements elicitation.
- 2) Ability to design a solution to a software problem
- 3) Ability to create UML diagrams, including class, object, use case, interaction, and state machines
- 4) Ability to identify various testing techniques and strategies and develop test cases for black box testing, white-box testing, and boundary testing
- 5) Demonstrate knowledge of the SDLC and associated activities
- 6) Develop use case diagrams and scenarios that address a given software problem
- 7) Develop solutions to a software problem
- 8) Demonstrate a knowledge of software development methodologies

### **COURSE REQUIREMENTS:**

1. Regular class attendance is the main requirement of this course.
2. Active class participation, this means you must spend some quality time preparing for your next class.
3. Programming assignments, homework, reports of hands-on labs, and deliverables must be turned in on time when they are due. Unfinished programs and non-working programs turned in on time will be graded; however, assignments not turned in before or on the due date will **NOT** be accepted. This means that you should start early to work on your programming assignments. Programs must be well-documented to be understood by a novice programmer.
4. Short quizzes may be given without prior notice and **there will be no making up of missed quizzes.**
5. One examinations and a final examination will be given. There will be **NO** make up for missed exams.
6. You have a responsibility and an obligation to prevent abuse and misuse of the university computer resources. Please read the UTC Computer Use Code of Conduct.
7. Individual extra credit assignments for the purpose of propping up a bad grade will **NOT** be given.

Notes taking is encouraged. Notes for the class can be found at the [bb4.utc.edu](http://bb4.utc.edu). You can also check the website of textbook for more resources.

## GRADE DISTRIBUTION:

90+ = A; 80-89 = B; 70-79 = C; 60-69 = D; below 60 = F.

Term Presentation and Project:	10%
Homework and Assignment	30%
Mid-term Exam	30%
Final	30%
Total	100%

## TENTATIVE SCHEDULE

Week	Topics
Week 1	Introduction and software processes
Week 2	Agile software development and requirement engineering
Week 3	System modeling and architectural design
Week 4	Design and implementation
Week 5	Software testing and software evolution
Week 6	Sociotechnical systems
Week 7	Dependability and security specification
Week 8	Dependability and security engineering
Week 9	Software reuse
Week 10	Component-based software engineering
Week 11	Service-oriented architecture
Week 12	Embedded software
Week 13	Aspect-oriented software engineering
Week 14	Configuration management
Week 15	Presentation

## TEXT BOOKS:

Required: Ian Sommerville, software engineering, 9<sup>th</sup> Edition, Pearson, 2010. ISBN: 978-0-13-703515-1.

## Course Website and Communication

You can access lecture notes, assignments, and your grades through Blackboard system. The blackboard system is used to communicate with students. Students can use email to communication with the instructor during the week. Emails received after 10pm or during weekend may not be replied. This includes the night before exams.

## Makeup/Late Policy

There will be no make-up tests. The final exam grade will replace an exam you miss. **Failure to take the final exam will result in failing**

**the course.** All assignments are to be turned in on or before the assigned due date. You must demonstrate that your lab or assignment is working properly. To verify you must have your lab assignments signed by the instructor. A 25% penalty will be assessed for late assignments for the first week. **No programming assignments will be accepted after the second late week and a grade of zero will be assigned for that assignment.**

## **Career Planning**

If you find that personal problems, career indecision, study and time management difficulties, etc. are adversely affecting your successful progress at UTC, please contact the Counseling and Career Planning Center at 425-4438.

## **\*UTC's Honor Code**

The UTC *Student Handbook* describes the Honor Code (pages 7 - 9), which includes the following examples of violations related to computer usage: (UTC Student Handbook page 7 paragraph B.2)

1. Making use of unauthorized assistance during an examination or in preparing a graded assignment
2. Plagiarism
3. Making unacknowledged use of another's computer program
4. Unauthorized use, or misuse, of the University's computing facilities such as:
  - Logging on to an account without the knowledge and permission of the owner
  - Changing, deleting, and adding to the programs, files and data without authorization of the owner
  - Theft of program data and machine resources
  - Attempts to thwart security of any computer system
  - Attempts to disrupt the normal operations of any computer system

In addition, I will not tolerate the use of cell phones in my class. If you have an emergency situation please let me know so accommodations can be made.

Any suspected Honor Code violation in this course will be forwarded to the Honor Court for action, and an F will be assigned for the course grade. All graded work in this course is subject to the Honor Code, including examinations, programming exercises, and any written work prepared for the course.

## **Important Dates**

Class begins	Jan. 6 (Monday)
Last day to withdraw before fee obligation	Jan. 12 (Sunday)

Last Day to Withdraw without a W	Jan. 19 (Sunday)
Martin Luther King holiday	Jan. 20 (Monday)
Midterm grade notifications	Feb. 17-21
Last Day to Withdraw with a W	March 9 (Sunday)
Spring Break	March 10-16
Spring Holiday	April 18
Last Day of Classes	April 21
Reading Day	April 22
Final Exam	Saturday, April 26: 10:30-
12:30 pm	