SEG 2105/Fall 2011

SEG 2105 INTRODUCTION TO SOFTWARE ENGINEERING

Principles of software engineering: Requirements, design and testing. Review of principles of object orientation. Object oriented analysis using UML. Frameworks and APIs. Introduction to the client-server architecture. Analysis, design and programming of simple servers and clients. Introduction to user interface technology. Prerequisite: ITI1121 or ITI1221

PROFESSOR:

Dr. Timothy C. Lethbridge **613 562-5800 ext: 6685**
Email: tcl@site.uottawa.ca Answers to non-personal email questions will be sent to the entire class, with the identity of the question-asker suppressed.
Office: SITE 5070. Office visits are welcome, but email for an appointment please.

TEXTBOOK AND OTHER SUPPORT MATERIAL:

Mandatory Text: **"Object Oriented Software Engineering: Practical Software Development Using UML and Java, 2nd Ed"** by T C. Lethbridge and R. Laganière (http://www.lloseng.com)
A DVD-ROM of lectures from a previous year is also available (go to dvclass.com).

Course website http://www.site.uottawa.ca/~tcl/seg2105/ .

COURSE OBJECTIVES:

When you complete this course you should be able to understand::

- The software engineering process, including requirements gathering, specification, and testing.
- Principles of object-oriented analysis and design, as well as software architecture (particularly the client-server architecture) and basic UI design.
- The basics of UML, the standard way of expressing requirements and design in software engineering.

IMPORTANT INFORMATION ABOUT UNIVERSITY RULES:

- As in all courses in the faculty, class attendance is mandatory. As per academic regulations, students who do not attend 80% of the class will not be allowed to write the final examinations.
- All components of the course (labs, assignments, etc.) must be fulfilled, otherwise students may receive EIN as a final mark (equivalent to F). This is also true for student repeating the course.
- All students must read and adhere to the Regulation on Academic Fraud (see http://web5.uottawa.ca/mcs-smc/academicintegrity/regulation.php)
  In particular, make sure you don't copy from other groups or students in assignments.

LECTURE AND ASSIGNMENT INITIAL PLAN (subject to change):

	Date	*Material to be covered*	*Assignments / Labs*
1	Fri Sep 9	Chap. 1 - Software and Software Engineering	
2	Tue Sep 13	Chap. 2 - basics of object-orientation)	*Lab 1 - java review*
3	Fri Sep 15	Chap. 2 - inheritance, polymorphism and review of key Java concepts)	
4	Tue Sep 20	Chap. 2 (remainder) and start of Chap. 3 (reuse, frameworks, &basic client-server concepts)	*Lab 2 - Simplechat 1*
5	Fri Sep 23	Chap. 3 (client-server architecture, network concepts, and networking in Java)	Asg1 due (lab 1 writeup) group 5p.m.
6	Tue Sep 27	Chap. 3 (Object Client-Server Framework)	*Lab 3 - Simplechat 2*
7	Fri Sep 30	Chap. 3 (remainder - SimpleChat), and start of Chapter 5 (Class diagrams) (skip chapter 4 until later)	
8	Tue Oct 4	Chap. 5 - Class Diagrams (sl. 1-19: diagramming classes, associations, reflexive associations, etc.)	*Lab 4 - Umple*
9	Fri Oct 7	Chap. 5 (sl. 19-26: abuses of generalization, aggregation + problem-solving on board E 5.18/19/20)	Asg2 - (basic OO) due 5 p.m.
10	Tue Oct 11	Chap. 5 (OCL, process for developing diagrams)	
11	Fri Oct 14	Chap. 5 (Airline System; identifying operations) plus start Chapter 4	Asg 3 due (lab writeup) due 5 p.m.
	Tue Oct 18	**Midterm: Material up to and including Oct 14**	
12	Fri Oct 21	Chap. 4 (sl. 9-22, plus discussion of example requirements in the book) Chap. 6 - Patterns - (start)	
		Study break Oct 24-28	
13	Tues Nov 1	Chap. 6 (sl. 7-24; General Hierarchy [Composite]; Player-Role; Singleton; Observer; Delegation)	Asg 4 due (UML) indiv due 5 p.m.
14	Fri Nov 4	Chap. 6 (Adapter; Facade; etc.) plus Chap. 7 (Focusing on Users and Their Tasks)	
	Tues Nov 8	TA will lead students in UML examples. Prof. away.	
15	Fri Nov 11	Chap. 7 (Use Cases and UI Design)	Asg 5 (requirements) group - due 5 p.m.
16	Tue Nov 15	Chap. 7 (Evaluating and Implementing UIs) and Chap. 8 (Interaction Diagrams)	
17	Fri Nov 18	Chapter 8 (State and Activity Diagrams)	
18	Tue Nov 22	Chapter 9 (Architecting and Designing Software – Design process, cohesion and coupling)	
19	Fri Nov 25	Chap. 9 (Design Principles and Software Architecture)	Asg 6 ch 8 ind due 5 p.m.
20	Tue Nov 29	Chap. 9 (Pipe-and-Filter; Design Docs) and Chap. 10 (Testing and Inspections)	*Demos in labs*
21	Fri Dec 2	Chap. 10 (continued)	Asg 7 project due 5 pm
22	Tue Dec 6	Chap. 10 (remainder) and Chap. 11 (SE Process Models, Cost Estimation, Team Building etc.)	

MARKING SCHEME:

Midterm Test : Worth 15% of final grade (or more, see below)
Final Exam: Worth 45% of final grade (or more, see below)
Assignments: Worth 40% of final grade (or less, see below)

Attendance will be taken at random times. One point will be deducted for every lecture or lab missed after the first two, unless you have a valid medical or compassionate excuse. This rule is necessary since you cannot learn well by just reading the notes, and also university rules require attendance.

The following are special calculations for people who do well on assignments, but prove on the midterm and/or exam that they didn't really know the material:

Exam component = (Midterm out of 15 + Final Exam out of 45) * 100/60
If you obtain a grade of **less than 50%** in the exam component, then assignments will not be counted. In other words, in order to pass the course, you must obtain at least 50% in the exam component.

Assignment component = (Assignments out of 40) * 100/40
The **maximum you can get in the assignment component** is 20% more than the exam component (i.e. Exam component * 1.2).

If you have a valid excuse to miss the midterm (e.g. medical), then the final exam mark will also be used to substitute for the midterm mark, in other words the weight of the final exam will become 60%.