

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>)

Course website <http://www.eecs.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 / red means changed)

	<i>Date</i>	<i>Material to be covered</i>	<i>Assignments Due 5 p.m.</i>	<i>Labs (Mon/Thurs)</i>
1	Wed Sep 3	Chap. 1 - Software and Software Engineering		
2	Fri Sep 5	Chap. 2 - Basics of object-orientation)		
3	Wed Sep 10	Chap. 2 - inheritance, polymorphism and review of key Java concepts)		8 th /9 th /11 th Lab 1 Java review
4	Fri Sep 12	Chap. 2 (remainder) and start of Chap. 3 (reuse, frameworks, & basic client-server concepts)		
5	Wed Sep 17	Chap. 3 (client-server architecture, network concepts, and networking in Java)		15 th /16 th /18 th Lab 2a Simplechat a
6	Fri Sep 19	Chap. 3 (Object Client-Server Framework)	Asg1 Lab 1 writeup (group) Due 5 pm Sep 21	
7	Wed Sep 24	Chap. 3 (remainder - SimpleChat), and start of Chapter 5 (Class diagrams) (skip Ch 4 for now)		22 nd /23 rd /25 th Lab 2b Simplechat b
8	Fri Sep 26	Chap. 5 - Class Diagrams (sl. 1-19: diagramming classes, associations, reflexive associations, etc.)	Asg2 - (basic OO) due 5 p.m. Sep 28	
9	Wed Oct 1 By video	** Professor away – Course online Manufacturing plant controller example.		29 th /30 th /2 nd Lab 3a Umple a
	Oct 3	NO CLASS – Work on Assignments	Asg3 - (lab 2 writeup) due 5 pm Oct 5	
10	Wed Oct 8	Chap. 5 (sl. 19-26: abuses of generalization, aggregation + problem-solving on board		6 th /7 th /9 th Lab 3b Umple b
11	Fri Oct 10	Chap. 5 (OCL, process for developing diagrams)		
		Thanksgiving – Monday Oct 13 Study break Oct 14-17	Asg 4 (UML) indiv due 5 pm Oct 20	
12	Wed Oct 22	Chap. 5 (Airline System; identifying operations) plus start Chapter 4		
	Fri Oct 24	Chap. 4 (sl. 9-22, plus discussion of example requirements in the book) Chap. 6 - Patterns		
	Sat Oct 25	Midterm: 9:00-10:20 a.m. Section A: STE G0103 and Section B: STE H0104		
13	Wed Oct 29	Chap. 6 (sl. 7-24; General Hierarchy [Composite]; Player-Role; Singleton; Observer; Delegation)		
14	Fri Oct 31	Chap. 6 (Adapter; Facade; etc.) plus Chap. 7 (Focusing on Users and Their Tasks)	Asg 5 (models) group - due 5 pm Nov 4	
15	Wed Nov 5	Chap. 7 (Use Cases and UI Design)		
16	Fri Nov 7	Chap. 7 (Evaluating and Implementing UIs) and Chap. 8 (Interaction Diagrams)		
17	Wed Nov 12	Chapter 8 (State and Activity Diagrams)		
18	Fri Nov 14	Chapter 9 (Architecting and Designing Software – Design process, cohesion/coupling)		
19	Wed Nov 19	Chap. 9 (Design Principles and Software Architecture)	Asg 6 ch 8 ind due 5pm Nov 16	
20	Fri Nov 21	Chap. 9 (Pipe-and-Filter; Design Docs) and Chap. 10 (Testing and Inspections)		
21	Wed Nov 26	Chap. 10 (continued)		Demos Nov 24/25/27
22	Fri Nov 28	Chap. 10 (remainder) and Chap. 11 (SE Process Models, Cost Estimation, Teams	Asg 7 project due Dec 1, 5 pm	

MARKING SCHEME:

Midterm Test : Worth 15% of final grade (or more, see below)

Final Exam: Worth 45% of final grade (or more, see below)

Participation: Worth 5% of the final grade. Based on answering questions in class at random times.

Assignments: Worth 35% of final grade (or less, see below)

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 35) * 100/35

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%.