SEG 2105/Fall 2018

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. Miguel A. Garzón 613 562-5800 ext: 2129
Email: mgarzon@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 5026B. 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 https://uottawa.brightspace.com/d2l/

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 INITIAL PLAN (subject to change / red means changed)

Date	*Material to be covered*
Wed Sep 5	Chap. 1 - Software and Software Engineering
Fri Sep 7	Chap. 2 - Basics of object-orientation)
Wed Sep 12	Chap. 2 - inheritance, polymorphism and review of key Java concepts)
Fri Sep 14	Chap. 2 (remainder) and start of Chap. 3 (reuse, frameworks, & basic client-server concepts)
Wed Sep 19	Chap. 3 (client-server architecture, network concepts, and networking in Java)
Fri Sep 21	Chap. 3 (Object Client-Server Framework)
Wed Sep 26	Chap. 3 (remainder - SimpleChat), and start of Chapter 5 (Class diagrams) (skip Ch 4 for now)
Fri Sept 28	Chap. 5 - Class Diagrams (sl. 1-19: diagramming classes, associations, reflexive associations, etc.)
Wed Oct 3	Manufacturing plant controller example.
Fri Oct 5	Library system example
Wed Oct 10	Chap. 5 (sl. 19-26: abuses of generalization, aggregation + problem-solving on board
Fri Oct 12	Chap. 5 (OCL, process for developing diagrams)
Wed Oct 18	Chap. 5 (Airline System; identifying operations) plus start Chapter 4
Fri Oct 19	Chap. 4 (sl. 9-22, plus discussion of example requirements in the book) Chap. 6 - Patterns
Sat Oct 20	Midterm: 9:00-10:30 a.m. Place TBD
	Study break Oct 23 to 29
Wed Oct 31	Chap. 6 (sl. 7-24; General Hierarchy [Composite]; Player-Role; Singleton; Observer; Delegation)
Fri Nov 2	Chap. 6 (Adapter; Facade; etc.) plus Chap. 7 (Focusing on Users and Their Tasks)
Wed Nov 7	Chap. 7 (Use Cases and UI Design)
Fri Nov 9	Chap. 7 (Evaluating and Implementing UIs) and Chap. 8 (Interaction Diagrams)
Wed Nov 14	Chapter 8 (State and Activity Diagrams)
Fri Nov 16	Chapter 9 (Architecting and Designing Software – Design process, cohesion/coupling)
Wed Nov 21	Chap. 9 (Design Principles and Software Architecture)
Fri Nov 23	Chap. 9 (Pipe-and-Filter; Design Docs) and Chap. 10 (Testing and Inspections)
Wed Nov 28	Chap. 10 (continued)
Fri Nov 30	Chap. 10 (remainder) and Chap. 11 (SE Process Models, Cost Estimation, Teams
Wed Dec 5	Review

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.

- **2% for attendance (if attendance in TopHat is greater than 75% then the student obtains 2%)**
- **3% for the work done in labs 3-6 (See Table below for details)**

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

- **20% for Project (4 deliverables)**
- **15% for 4 assignments**

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%.

The following Table shows the weight distribution for all assignments and project deliverables:

Grade Item	Weight
Assignment 1	2.5%
Assignment 2	6%
Assignment 3	3.5%
Assignment 4	3%
Project Deliverable 1	2%
Project Deliverable 2	5%
Project Deliverable 3	5%
Project Deliverable 4	8%
Lab 3 - 6 Bonus is obtained if the app in labs 4-6 is completed and showed to TA	0.5% + 0.25% Bonus