

# SEG 4110/Fall 2011

## SEG4110 ADVANCED SOFTWARE DESIGN AND REENGINEERING

Aspect-oriented design and metaprogramming. Model driven architecture. Reverse engineering, program understanding, re-engineering, automated program transformation and refactoring. Other advanced techniques for design and generation of software systems. Prerequisite: SEG3202

### **PROFESSOR:**

**Dr. Timothy C. Lethbridge**

**562-5800 ext: 6685**

**Email:** [tcl@site.uottawa.ca](mailto: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. You are responsible for reading your email since the messages might announce important changes.

**Office:** SITE 5070. Office visits are welcome, but email for an appointment please.

### **SUPPORT MATERIAL:**

Slides and web links will be posted on the course website from time to time.

<http://www.site.uottawa.ca/~tcl/seg4110/>

### **COURSE OBJECTIVES:**

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

- A wide variety of architectures and technologies available to design and implement software
- Basic principles of maintenance and re-engineering.

### **LECTURE AND ASSIGNMENT INITIAL PLAN (subject to change):**

**I intend to cover the following topics, but there may be adjustments**

- |                                                                 |                                                                    |
|-----------------------------------------------------------------|--------------------------------------------------------------------|
| A. UML Review and Design Examples                               | L. Garbage Collection                                              |
| B. UML Extension Mechanisms                                     | M. Security                                                        |
| C. Advanced OO Modelling, including a new language called Umple | N. Ruby on Rails, a language and framework for generating websites |
| D. Metamodelling                                                | O. Object-Relational Mapping                                       |
| E. Formal Approaches: OCL                                       | P. Ajax                                                            |
| F. Java Collections Framework                                   | Q. Component Frameworks                                            |
| G. Aspect Oriented Programming in Aspect J                      | R. Basics of Software evolution and maintenance                    |
| H. Java Reflection                                              | S. Program Analysis                                                |
| I. Review/Introduction to C++                                   | T. Refactoring                                                     |
| J. C++ Standard Template Library                                |                                                                    |
| K. C-Sharp basics                                               |                                                                    |

I am more interested in you learning general design principles, than cramming in a maximum amount of knowledge, so I may shorten or drop one or more of the above topics if needed. I may also add a new topic. The slides are subject to change, so don't print them all out right away.

**MIDTERM:** Tentatively scheduled for **Web Oct 19th** in class

### **IMPORTANT INFORMATION ABOUT UNIVERSITY RULES:**

- As in all courses in the faculty, class attendance is mandatory. 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).
- All students must read and adhere to the Regulation on Academic Fraud (see <http://web5.uottawa.ca/mcs-smc/academicintegrity/regulation.php> )

## **LABS Mondays 8:30-10:00 STE 0130 (subject to adjustment):**

TA for the labs: To be determined. Note, the class follows right afterwards. Please leave the lab in time to arrive in the class.

Sep 12	No lab scheduled	
Sep 19	Closed Lab 1	Umple
Sep 26	Closed Lab 2 (1/2)	Rational Software Modeler
Oct 3	Closed Lab 2 (2/2)	Rational Software Modeler
Oct 10	No lab - Thanksgiving	
Oct 17	Closed Lab 3 (1/2)	Aspect-oriented Modeling
Oct 24	No lab (study break)	
Oct 31	Closed Lab 3 (2/2)	Aspect-oriented Modeling
Nov 7	Closed Lab 4	C++ Standard Template Library
Nov 14	Closed Lab 5	Ruby on Rails
Nov 21, 28	Project Help (by appointment)	
Dec 5, 7	Project demonstrations. Note that Dec 7 follows Monday schedule.	

## **MARKING SCHEME:**

**Midterm Test: Worth 10% of final grade**

**Final Exam: Worth 45% of final grade**

**Assignments and labs: Worth 45% of final grade**

If you have a valid excuse to miss the midterm (e.g. medical), then the final exam mark will also be used as the midterm mark.

## **ASSIGNMENTS AND MARK BREAKDOWN:**

**8% Lab work.** Part of the grade is for attendance, and part for completing tasks and handing them in.

**12% . UML Design Assignment** UML, OCL and Umple: Creating a model and generating code. Done individually.

**25% Technology project.** Done in groups of 2

You will build a small application using one of the technologies we discuss in class or some other technology approved by the professor. Before proceeding, you must form groups and check with the professor. Only one project per technology, so first-come, first-served.

**3%: Proposal.** You must write about the technology you are planning to use, and describe a small technology-demonstration project you will do with this. The TA will give you feedback on the feasibility of your work, and grade you on the extent to which you have demonstrated that you understand the technology you will be using. 5-6 single-spaced pages expected.

**5%: Initial report:** You will complete the first iteration of your project, complete with a user guide, and test cases, and then pass this to another group. The TA will primarily evaluate the quality of your user guide and testcases. 7-10 single-spaced pages expected.

**4%: Testing report of another group's work.** You will report on your experiences using the other group's project.

**8% Presentation in class of your own work.** You will spend 15 minutes describing how you applied the technology, and 15 minutes demonstrating your work.

**10%: Final report of your own work:** Code + report on changes you made in response to the testing.