SEG 4110/Fall 2015

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 Answers to non-personal email questions will be sent to the entire class. You are responsible for reading your email.
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/
Model-Driven Software Engineering in Practice (downloadable)
https://www.sites.google.com/site/mdsebook/

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 topics such as the following, subject to existing knowledge analysis.

UML Review and Design Examples
UML Extension Mechanisms
Domain Specific Modeling
Advanced OO Modelling, including a new language called Umple
Metamodelling
Formal Approaches: OCL
Java Collections Framework
Aspect Oriented Programming in Aspect J
Java Reflection
Review/Introduction to C++
C++ Standard Template Library
C-Sharp basics
Garbage Collection
Security
Ruby on Rails, a language and framework for generating websites
Object-Relational Mapping
Ajax
Component Frameworks
Basics of Software evolution and maintenance
Program Analysis
Refactoring

I am more interested in you learning general design principles, than cramming in a maximum amount of knowledge. The slides are subject to change, so don't print them all out right away.

MIDTERM: In class; date to be determined

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 TBD:

TA to be determined.

Potential bab topics: Aspect-Oriented Modeling, C++ Standard Template Library, Ruby on Rails

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 and assignment work.
12%. Design Assignment UML, OCL, Umple, DSL, Design rationale: 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.