ENGG*4450: Large-Scale Software Architecture Engineering
Fall 2015

UNIVERSITY of GUELPH
School of Engineering

(Revision 2: September 14, 2015

---

1 Instructional Support

1.1 Instructor

Instructor:	Graham Taylor, Ph.D.
Office:	RICH 3515
Email:	gwtaylor@uoguelph.ca
Office hours:	Monday 16:00 - 17:00

1.2 Lab Technician

Technician:	Joel Best, Information Technology Manager
Office:	RICH 3501
Email:	jbest@uoguelph.ca

1.3 Teaching Assistants

GTA	Email	Office Hours
Ryan Pattison	rpattiso@uoguelph.ca	GTA can be consulted during lab hours.

---

2 Learning Resources

2.1 Course Website

Course material, news, announcements, and grades will be regularly posted to the ENGG*4450 CourseLink site. You are responsible for checking the site regularly.

2.2 Required Resources

There is no required textbook for the course.

2.3 Recommended Resources

You will find it useful to consult a UML reference book from time to time. The following book is recommended:

Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", 3rd edition, Addison-Wesley, 2003.

The following text is also a useful reference (for those who like textbooks):

Bernd Bruegge and Allen H. Dutoit, "Object-Oriented Software Engineering", 3rd edition, Prentice Hall, 2010.

The above books are on reserve at the library. You will need to refer to other books and readings throughout the course – we will provide pointers as needed.

2.4 Additional Resources

Lecture Information: All of the lecture notes will be posted on the web page as the term progresses. You are encouraged to print out a copy of the lecture notes before class so that you can annotate them during the lecture.

Lab Information: The handouts for all the lab sessions will be provided within the lab section. All types of resources regarding tutorials, links to web pages, etc. can be found in this section.

Assignments: Download the assignments according to the schedule given in this handout.

Exams: Some final exams of previous years are posted as samples. The solutions are also posted for your convenience.

Miscellaneous Information: Other information related to Large-scale Software Architecture Engineering will be posted on the course website.

2.5 Communication & Email Policy

Please use lectures and lab help sessions as your main opportunity to ask questions about the course. Major announcements and/or changes will be posted to the course website. **It is your responsibility to check the course website regularly.**

Electronic communication should be limited to the **course forum**, however topics of a personal and confidential nature (e.g. marks) should be emailed to the instructor: gwtaylor@uoguelph.ca. As per university regulations, all students are required to check their *mail.uoguelph.ca* e-mail account regularly: e-mail is the official route of communication between the University and its students.

3 ASSESSMENT

3.1 Dates and Distribution

Assignments: 40%

A1: Reverse Engineering & Design Recovery (5%), due Oct 7, in class
A2: Implementing Change Requests (10%), due Oct 28, in class
A3: Requirements Analysis (10%), due Nov 18, in class
A4: Build and Test a New Feature (15%), due Dec 9, by PDF to CourseLink dropbox

You will work in teams of approximately 6-7 students for all assignments. Teams will be announced in the first or second week of the term. Each team will submit a single report for each assignment. All members of a team will receive the same grade for the assignment, except in exceptional circumstances at the discretion of the instructor. Detailed instruction on the content of each assignment will be distributed during the term.

Lab Test: 20%

The lab test will be held Nov 3 in the regular lab session. It will test all material covered in lectures and labs to-date.
See section 5.3 below for more details on labs.

Final Exam: 40%

Thu Dec 17, 19:00 - 21:00, Room TBA on Webadvisor

3.2 Course Grading Policies

Passing Grade: The passing grade is 50%.

Missed Assessments: If you are unable to meet an in-course requirement due to medical, psychological, or compassionate reasons, please email the course instructor. See the undergraduate calendar for information on regulations and procedures for Academic Consideration:
http://www.uoguelph.ca/registrar/calendars/undergraduate/current/c08/c08-ac.shtml

Accommodation of Religious Obligations: If you are unable to meet an in-course requirement due to religious obligations, please email the course instructor. See the undergraduate calendar for information on regulations and procedures for Academic Consideration of Religious Obligations:
http://www.uoguelph.ca/registrar/calendars/undergraduate/current/c08/c08-accomrelig.shtml

Missed Lab Test: If you miss a test due to grounds for granting academic consideration or religious accommodation, the weight of any missed test will be added to the final exam weight. There will be no makeup lab tests.

Lab Work: You must attend and complete all laboratories. If you miss a laboratory due to grounds for granting academic consideration or religious accommodation, arrangements must be made with the teaching assistant to complete a makeup lab.

Late Assignments: Due dates for the assignments are firm. Assignments must be submitted in person, within ten minutes of the start of the lecture on the specified date (i.e. by 11:40). Late submissions of assignments will not be accepted.

Clarification About Grades: If you have questions about the grade your assignment received, please ask your TA. However, all requests for re-marking must be made to the instructor. Any item that is re-marked will be re-marked entirely. Therefore it is strongly suggested that you thoroughly review your entire document before making a re-marking request. Re-marking requests will not be honoured more than one week after the document has been returned.

---

4 AIMS, OBJECTIVES & GRADUATE ATTRIBUTES

4.1 Calendar Description

This course introduces the students to the analysis, synthesis and design of large-scale software systems at the architectural level. This is in contrast to the algorithmic and data structure viewpoint of most software systems. Large-scale software systems are complex, execute on many processors, under different operating systems, use a particular or many language(s) of implementation, and typically rely on system layers, network connectivity, messaging and data management and hardware interfacing. The material covered includes architectural styles, case studies, architectural design techniques, formal models, specifications and architectural design tools. The laboratory sessions will expose the students to analyzing and redesigning an existing large-scale software system. *Prerequisite(s):* (CIS*2420 or CIS*2520), ENGG*2100

4.2 Course Aims

This course caps a series of several software development and programming courses that started with CIS*1500 and included courses on object oriented design, algorithms and data structures. This course introduces software engineering processes and tools to deal with the complexity involved in designing, implementing and testing of large scale software systems that may involve tens or hundreds of developers. We will emphasize open-source and distributed development models.

"Software development is no longer bound by time zones or national borders. Projects of all kinds – academic, commercial, and open sourcemay have their GUI designers in Boston, their database team in Bangalore, and their testers in Budapest and Buenos Aires. Working effectively in such teams is challenging: it requires strong communication skills, and makes proper use of coordination tools such as version control and ticketing systems more important than ever." – Undergraduate Capstone Open Source Projects, http://ucosp.ca/about/

4.3 Learning Objectives

At the successful completion of this course, the student will have demonstrated the ability to:

1. Analyze client requirements and formulate use cases.
2. Reverse engineer a design through code inspection and reverse engineering tools.
3. Produce design specifications according to standards (e.g. UML).
4. Describe and compare different software development methodologies.

5. Master version control and concepts such as branching and merging.
6. Apply verification and validation strategies to software development.
7. Concisely and articulately communicate through written documentation for software projects.
8. Analyze risk in software projects.
9. Work effectively as part of a medium to large development team.

4.4 Graduate Attributes

Successfully completing this course will contribute to the following CEAB Graduate Attributes:

Graduate Attribute	Learning Objectives	Assessment
1. Knowledge Base for Engineering	2, 5	Assignments, Exam
2. Problem Analysis	1, 3	Assignments, Lab Test, Exam
3. Investigation	1, 2, 4	-
4. Design	1, 2, 3, 6, 7	Assignments, Exam
5. Use of Engineering Tools	2, 3, 5, 6	Assignments, Lab Test
6. Communication	1, 3, 7, 9	Assignments
7. Individual and Teamwork	9	Assignments
8. Professionalism	-	-
9. Impact of Engineering on Society and the Environment	1, 6, 8	-
10. Ethics and Equity	-	-
11. Environment, Society, Business, & Project Management	6	Assignments, Exam
12. Life-Long Learning	-	-

4.5 Instructor's Role and Responsibility to Students

The instructor's role is to develop and deliver course material in ways that facilitate learning for a variety of students. Selected lecture notes will be made available to students on CourseLink/D2L but these are not intended to be stand-alone course notes. During lectures, the instructor will expand and explain the content of notes and provide example problems that supplement posted notes. Scheduled classes will be the principal venue to provide information and feedback for tests and project.

4.6 Students' Learning Responsibilities

Students are expected to take advantage of the learning opportunities provided during lectures and tutorials. Students, especially those having difficulty with the course content, should also make use of other resources recommended by the instructor. Students who do (or may) fall behind due to illness, work, or extra-curricular activities are advised to keep the instructor informed. This will allow the instructor to recommend extra resources in a timely manner and/or provide consideration if appropriate.

4.7 Relationships with other Courses & Labs

Previous Courses:

CIS*2520: Data structures & algorithms.

ENGG*2100: Design practices.

Follow-on Courses:

ENGG*4120 or ENGG*4170: Collaborative design project.

---

5 Teaching and Learning Activities

5.1 Timetable

Lectures:

Monday	11:30 - 12:50	ALEX 117
Wednesday	11:30 - 12:50	ALEX 117

Laboratory:

Monday	08:30 - 10:20	THRN 1313

5.2 Lecture Schedule

Lectures	Lecture Topics	Learning Objectives
1	Orientation	-
2-3	Modeling & UML	4
4	Modeling Software Behaviour	4
5	Showing the Architecture	4
6	Software Re-Engineering	1, 3, 5
7	Software Processes	1, 3
8	Requirements Elicitation & Use-Case Driven Design	2, 3, 4, 6, 8
9	Estimation and Prioritization	4, 5
10	Risk Management	8, 9, 11
11	Project Management	6, 7
12	Requirements Analysis	2, 3
13	From Requirements to Design	3, 4
14	Robustness Analysis	4
15	Verification & Validation	8
16-18	Testing	5
19	Static Analysis	5
20	Managing Software Teams	4, 5
21	Mobile App Development	6, 7
22	Software Quality	9, 10
23	DevOps	5
24	Course Review	-

5.3 Lab Schedule

There are several labs in the course meant to familiarize students with software engineering tools. The labs will be followed by a lab exam which will test proficiency in software use as well as practical application of concepts covered in lectures.

Week	Activity
2	Introduction to lab
3	Object-oriented analysis and design
4	Assignment 1 help session
5	Version control with Git
6	Thanksgiving - no lab
7	Assignment 2 help session
8	Code review
9	Lab test
10	Assignment 3 help session
11	Automated testing
12	Catch-up help session
13	Assignment 4 help session

5.4 Other Important Dates

Thursday, 10 September 2015: First class

Monday, 12 October 2015: Thanksgiving holiday

Tuesday, 13 October 2015: Fall Study Break Day – no classes scheduled

Friday, 6 November 2015: Last day to drop – 40th class

Friday, 4 December 2015: Last day of class (Monday schedule in effect)

---

6 LAB SAFETY

Safety is critically important to the School and is the responsibility of all members of the School: faculty, staff and students. As a student in a lab course you are responsible for taking all reasonable safety precautions and following the lab safety rules specific to the lab you are working in. In addition, you are responsible for reporting all safety issues to the laboratory supervisor, GTA or faculty responsible.

7 Academic Misconduct

The University of Guelph is committed to upholding the highest standards of academic integrity and it is the responsibility of all members of the University community faculty, staff, and students to be aware of what constitutes academic misconduct and to do as much as possible to prevent academic offences from occurring. University of Guelph students have the responsibility of abiding by the University's policy on academic misconduct regardless of their location of study; faculty, staff and students have the responsibility of supporting an environment that discourages misconduct. Students need to remain aware that instructors have access to and the right to use electronic and other means of detection.

Please note: Whether or not a student intended to commit academic misconduct is not relevant for a finding of guilt. Hurried or careless submission of assignments does not excuse students from responsibility for verifying the academic integrity of their work before submitting it. Students who are in any doubt as to whether an action on their part could be construed as an academic offence should consult with a faculty member.

7.1 Resources

The Academic Misconduct Policy is detailed in the Undergraduate Calendar:
http://www.uoguelph.ca/registrar/calendars/undergraduate/current/c08/c08-amisconduct.shtml

A tutorial on Academic Misconduct produced by the Learning Commons can be found at:
http://www.academicintegrity.uoguelph.ca/

Please also review the section on Academic Misconduct in your Engineering Program Guide.

The School of Engineering has adopted a Code of Ethics that can be found at:
http://www.uoguelph.ca/engineering/undergrad-counselling-ethics

8 Accessibility

The University of Guelph is committed to creating a barrier-free environment. Providing services for students is a shared responsibility among students, faculty and administrators. This relationship is based on respect of individual rights, the dignity of the individual and the University community's shared commitment to an open and supportive learning environment. Students requiring service or accommodation, whether due to an identified, ongoing disability for a short-term disability should contact the Centre for Students with Disabilities as soon as possible

For more information, contact CSD at 519-824-4120 ext. 56208 or email csd@uoguelph.ca or visit the website: http://www.csd.uoguelph.ca/csd/.

9 Recording of Materials

Presentations which are made in relation to course work–including lectures–cannot be recorded or copied without the permission of the presenter, whether the instructor, classmate or guest lecturer. Material recorded with permission is restricted to use for that course unless further permission is granted.

10 Resources

The Academic Calendars are the source of information about the University of Guelphs procedures, policies and regulations which apply to undergraduate, graduate and diploma programs:
http://www.uoguelph.ca/registrar/calendars/index.cfm?index.