ENGINEERING 5895: Software Design

Instructor **Andrew Vardy**
E-mail **av at mun dot ca**
Phone **864-4521**
Office Location **EN-2018**
Office Hours **Mondays and Wednesdays from 1:00 – 2:00**

TA **Hafez Seliem**
TA's e-mail **hms117 at mun.ca**
TA's office **EN-4031**

Website **www.cs.mun.ca/~av/courses/5895-current**

Communication

CALENDAR ENTRY:

ENGI 5895 Software Design examines the development process: requirements analysis, design, iterative development, design documentation; an introduction to the Unified Modelling Language: use cases, class diagrams and sequence diagrams; an introduction to software design patterns: creational patterns, structural patterns and behavioral patterns; object oriented, modular decomposition. The course includes a major design project.

LC: 25 lecture hours per semester
LH: six 3-hour sessions per semester
OR: meetings with project supervisor as required
PR: ENGI 4892

LAB EXPERIENCE: at least four 3-hour sessions per semester

CREDIT VALUE: 3 credit hours

COURSE TYPE: compulsory (Computer Engineering)

CONTENT CATEGORIES: (expressed as %, no category can be 0 < c < 25)

Math	Natural science	Complementary Studies	Engineering Science	Engineering Design
				100

COURSE DESCRIPTION:

In this course, students will combine technical mastery with creativity to create complete software applications. The student will learn about software design at the requirements, architectural, pattern, and class levels. There will be a major project completed in groups of 2-3.

SCHEDULE:

LECTURE:	Mon, Wed, Fri 12:00 – 12:50	Room: EN-1004	
MEETING / LAB:	Thur 9:00 – 12:00	Room: EN-1038B/~~4035~~ 1000	

RESOURCES:

TEXT BOOK

- ***THERE IS NO REQUIRED TEXT BOOK FOR THIS COURSE***

REFERENCES

- Robert C. Martin, Agile Software Development, Prentice Hall, 2003.
- Martin Fowler, UML Distilled, 3rd ed, Addison Wesley, 2004. (Available electronically from the library)
- Erich Gamma, Richard Helm, Ralph Johnson, and Jon Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley, 1995.
- Craig Larman, Applying UML and Patterns, Prentice Hall, 2005.
- Eckel, B., Thinking in Java, (3rd edition available for free online at http://www.
- mindview.net/Books/TIJ).

MAJOR TOPICS:

- Java programming
- UML
- Software Engineering: products and processes
- Software Design principles
- Object-Oriented Design patterns
- Application of all the above to a major project

LEARNING OUTCOMES:

Course Level Graduate Attribute Focus: Des-D, PA-D, Tools-D
1 A knowledge base for engineering 2 Problem analysis 3 Investigation 4 Design 5 Use of engineering tools

	LEARNING OUTCOMES At the appropriate milestones, the student will be expected to be able to:	**GRADUATE ATTRIBUTES. LEVEL OF COMPETENCE**	**Methods of Assessment**
1	Understand and document software designs using appropriate linguistic and visual formalisms and tools. Short version: Document their designs well.	1.A Knowledge base – Applied 4.A Design – Applied. 5.A Use of tools – Applied. 7.A Communication – Applied.	Project design documents. Presentations, Exam.
2	Explain the basics of assertion-based design (design by contract).	1.D Knowledge base – Developed. 4.D Design – Developed.	Exam.
3	Apply the principles of object-oriented design.	1.3 Knowledge base – Applied. 4.3 Design – Applied.	Project design documents and code. Presentations. Exam.
4	Apply major object-oriented design patterns and show familiar with others.	1.2 Knowledge base – Developed. 4.2 Design – Developed.	Project design documents. Exam.
5	Manage a modest sized software design project.	6.D Individual and team work – Developed. 7.D Communication – Developed. 11.I Economics and project management – Introduced.	Project meetings and design documents.
6	Understand the advantages of sound modularization and well-defined interfaces.	1.D Knowledge base – Developed. 4.D Design – Developed.	Project meetings and design documents. Exam.

*Each Graduate Attribute for each learning outcome is rated at a Content Instructional Level of I=Introduced, D=Developed, or A=Applied.

See www.mun.ca/engineering/undergrad/graduateattributes.pdf for definitions on the 12 Graduate Attributes and the Content Instructional Levels.

ASSESSMENT / TENTATIVE SCHEDULE:

Assignments (2)	15%	Wed Jan 24, Wed Jan 31
Lab	0%	Thurs Jan 25
Project	60%	
	0%	Thurs Feb 1 (Project Idea Discussion)
	2%	Fri Feb 2 (Project Proposal)
	0%	Thurs Feb 8 (Proposal Review Meeting)
	10%	Thurs Feb 15 (Initial Design Presentation)
	10%	Mon Feb 26 (Initial Design Document)
	10%	Thurs Mar 8 (First Iteration Demo)
	10%	Wed Mar 14 (Second Iteration Report)
	0%	Thurs Mar 22 (Progress Meeting)
	13%	Thurs Mar 29 (Final Demo)
	5%	Mon Apr 2 (Final Code Submission)
Mid-term Exams (2)	25%	Mon Feb 12, Mon Mar 12

The project is a group project to be completed in teams of 2 - 3. Note that the performance expectations for groups of 3 will be greater than for groups of 2.

There will be a **peer review assessment** that students will be asked to submit for major project deliverables (e.g. the design documents, presentations, and demos). If the contributions of the 2 (or 3) students are significantly unbalanced, then marks may be adjusted accordingly. It is therefore important for all team members to strive towards an equal contribution of efforts.

LAB SAFETY:

Students are expected to demonstrate awareness of, and personal accountability for, safe laboratory conduct. Appropriate personal protective equipment (PPE) must be worn (e.g. steel-toed shoes, safety glasses, etc.) and safe work practices must be followed as indicated for individual laboratories, materials and equipment. Students will immediately report any concerns regarding safety to the teaching assistant, staff technologist, and professor.

ACADEMIC INTEGRITY AND PROFESSIONAL CONDUCT:

Students are expected to conduct themselves in all aspects of the course at the highest level of academic integrity. Any student found to commit academic misconduct will be dealt with according to the Faculty and University practices. More information is available at http://www.mun.ca/engineering/undergrad/academicintegrity.php

Students are encouraged to consult the Faculty of Engineering and Applied Science Student Code of Conduct at http://www.mun.ca/engineering/undergrad/academicintegrity.php and Memorial University's Code of Student Conduct at http://www.mun.ca/student/conduct/.

INCLUSION AND EQUITY:

Students who require accommodations are encouraged to contact the Glenn Roy Blundon Centre, http://www.mun.ca/blundon/about/index.php. The mission of the Blundon Centre is to provide and co-ordinate programs and services that enable students with disabilities to maximize their educational potential and to increase awareness of inclusive values among all members of the university community.

The university experience is enriched by the diversity of viewpoints, values, and backgrounds that each class participant possesses. In order for this course to encourage as much insightful and comprehensive discussion among class participants as possible, there is an expectation that dialogue will be collegial and respectful across disciplinary, cultural, and personal boundaries.

STUDENT ASSISTANCE: Student Affairs and Services offers help and support in a variety of areas, both academic and personal. More information can be found at www.mun.ca/student.

ADDITIONAL INFORMATION:

This course is focussed around a major project, which is to be completed by groups of 2-3 students. To allow more time to complete the project, lectures will end prior to the second mid-term exam on March 12. There is no final exam.

The assignments are to be completed individually or in pairs. You are not required to work with the same people for all assignments and the project. However, once the project teams are formed, you should do your best to work collaboratively with your teammates for the duration of the term.

The purpose of the lab is to introduce the Visual Paradigm CASE tool. We will be using the Java language for the assignments and project.