UNIVERSITY OF GUELPH
School of Computer Science
COURSE OUTLINE

Course Code: CIS*4250	**Course Title:** Software Design V	**Date of Offering:** Fall 2016

Instructor:	**Office:**	Reynolds 311
Dr. Deborah Stacey	**Extension:**	52250
	Email:	dastacey@uoguelph.ca
Teaching Assistants: *None*	**Hours:**	*To Be Posted*

Calendar Description: This is a capstone course which applies the knowledge gained from the previous Software Design courses to a large team project. The course has an applied focus and will involve software design and development experiences in teams, a literacy component, and the use of software development tools. *Prerequisites*: CIS*2750, CIS*3260, CIS*3750	**Topics:** • Project Management • Agile Software Development • Lean Software
Class Information: Lectures Tuesday / Thursday 8:30-11:20, REYN 114	**Seminar Information**

Required Texts:

- *None but a reading list will be provided on the course Moodle site.*

Method of Evaluation:

Course Element	**Date**	**Weight**
1. *Project Deliverables*		
• Specifications and Business Plan	See Page 3 for details.	15%
• Design Documents		20%
• Testing Documents and Test Suite		15%
• Product Demo		15%
• Final Product		25%
2. *Individual Deliverable*		
• Project post mortem		10%
Final Exam	None	

Grading Policies	The student must submit the individual deliverable and achieve positive project evaluations from their team members to pass the course. If the project evaluations are not positive the student will be warned immediately so that positive actions can be taken.

Website: *Moodle*

UNIVERSITY OF GUELPH
School of Computer Science
COURSE OUTLINE

ACADEMIC INTEGRITY

The University of Guelph is committed to the highest standards of academic integrity and honesty. Students are expected to be familiar with these standards, and must abide by the applicable policies (see Section VIII of the Undergraduate Calendar on "Academic Misconduct" found on **http://www.uoguelph.ca/registrar/calendars/undergraduate/current/c08/c08-amisconduct.shtml**)

For educational purposes, instructors impose conditions on assignments that may limit students' permission to collaborate with others or to utilize external sources (including, but not limited to, software, data, images, text, etc.). Any permitted utilization must be done with proper references. Instructors may use automated tools to detect possible cases of plagiarism. Work that shows significant unnatural similarity, or that appears to be copied from unacknowledged sources, will be investigated as potential academic misconduct. "Aiding and abetting" is also a punishable offence, and students must be careful not to help others commit offences by giving out their files or allowing others to access their computer accounts. ***Consider yourself warned.***

ACCEPTABLE USE POLICY

Please read the complete University of Guelph policy found on **https://www.uoguelph.ca/cio/content/aup-acceptable-use-policy**.

E-MAIL POLICY

Students should include their **name** and **course number** in every email, *e.g.* Joe Smith: CIS*4250, since instructors are often involved in teaching more than one course per term. To comply with university privacy policy, all emails should be sent from your ***mail.uoguelph.ca*** account (not from hotmail.com or any other non-UoG host). All students are responsible for reading their ***uoguelph*** email and therefore should maintain their accounts, *i.e.* disk quotes should be monitored so that email is not rejected due to lack of space. On the subject line of your email please include the course number, e.g. "*Subject*: CIS*4250 – Question about the demo"

UNIVERSITY OF GUELPH
School of Computer Science
COURSE OUTLINE

Evaluation Schedule

Activity	Due Date	Weight
Specification Document		**10**
- Version 1	Sep 23	4
- Version 2	Oct 14	4
- Final Update	Dec 1	2
Business and Impact Document		**5**
- Version 1	Oct 28	2
- Final Update	Dec 1	3
Design Document		**20**
- Version 1	Sep 30	6
- Version 2	Oct 21	8
- Final Update	Dec 1	6
Implementation Document		**6**
- Version 1	Oct 7	2
- Version 2	Nov 4	2
- Final Update	Dec 1	2
Implementation – System		**19**
- Builds (8 X 0.5)	*	4
- System	Dec 1	15
Testing Document		**10**
- Version 1	Oct 7	2
- Version 2	Oct 21	2
- Version 3	Nov 4	2
- Version 4	Nov 18	2
- Final Update	Dec 1	2
Testing Suite and System	Dec 1	**5**
Product Features		**15**
- Demo	**	10
- Documents (manual, online help, etc.)	Dec 1	5
Post Mortem (individual)	Dec 1	**10**

*
Builds are to done every Friday from Week 4 (Oct 7) to Week 11 (Nov 25)

**
Demos will be scheduled for each team in the last two weeks of class (Nov 22 – Nov 29)