UNIVERSITY OF SASKATCHEWAN

COURSE SYLLABUS

COURSE TITLE:	Intermediate Software Engineering		
COURSE CODE:	370	**TERM:**	Fall 2014-15
COURSE CREDITS:	3	**DELIVERY:**	MWF
CLASS SECTION: **CLASS LOCATION:** **CLASS TIME:** **WEBSITE:**	 Arts Building 214 10:30 am –11:20 am moodle.cs.usask.ca/Dropbox.com	**START DATE:** **LAB LOCATION:** **LAB TIME:**	Sep 3, 2014 Thorvaldson Building 159 **4:00 pm –5:20 pm**

Course Description

Principles and techniques for developing software combined with the practical experience of creating a mid-size software system as a member of a software development team. Includes: teamwork; projects, planning and process; users and requirements; use cases; modeling; quality; software architecture; testing; GUI design, design principles, patterns and implementation; ethics; professionalism. Topics include:

1. Course Overview and Project Description
2. Projects: Teamwork, Planning, Process, Risk, Agility, Quality, Technology
3. Users and requirements, use cases, design
4. Software Modelling, GUI Design, Implementation
5. Interactive Systems Design and implementation
6. Design principles, patterns and implementation
7. Advanced design, design patterns, refactoring
8. Inspection and Testing
9. Software Architecture
10. Project Presentations
11. Software Maintenance and Evolution

12. Professionalism and ethics, Course Review

Prerequisites

CMPT 214 and 250.
Note: As of September 2007, course prerequisites will be: CMPT 214 and 270.3

September 2007: CMPT 250.6 will be replaced by 270.3 and 280.3.
Students with credit for 250.6 do not need to take 270.3.

Learning Outcomes

By the completion of this course, students will be expected to learn the following objectives:

1. The students are expected to learn of how to think a real world problem as a software development problem.
2. The students are expected to learn of how to analyze a software development problem, and then design the problem in software development context using different modeling languages (e.g., UML). The students should be capable of providing in-depth analysis of the subject problem for the purpose of better design.
3. Once a tentative design is reached, the students are then expected to apply advanced software engineering techniques and principles both to improve the analysis and the design as outlined in the course syllabus.
4. The students are expected to apply all those from Step 2 and 3 to mid-size projects and implement first prototypes.
5. Based on further advanced techniques of software design and analysis (e.g., design patterns and refactoring techniques), the students then refine their implementations to make better software systems.
6. The students then learn advanced techniques of software testing and apply to their developed projects.
7. The students are expected to learn of how to work in a group, even in different challenging circumstances such as scheduling issues of the group mates.
8. The students are expected to learn the usability issues of the developed software and then they also refine the usability of the software systems they developed.

Information on literal descriptors for grading at the University of Saskatchewan can be found at: http://students.usask.ca/current/academics/grades/grading-system.php

Please note: There are different literal descriptors for undergraduate and graduate students.

More information on the Academic Courses Policy on course delivery, examinations and assessment of student learning can be found at:

http://policies.usask.ca/policies/academic-affairs/academic-courses.php

The University of Saskatchewan Learning Charter is intended to define aspirations about the learning experience that the University aims to provide, and the roles to be played in realizing these aspirations by students, instructors and the institution. A copy of the Learning Charter can be found at: http://www.usask.ca/university_secretary/LearningCharter.pdf

University of Saskatchewan Grading System (for undergraduate courses)

Exceptional (90-100) A superior performance with consistent evidence of

- a comprehensive, incisive grasp of the subject matter;
- an ability to make insightful critical evaluation of the material given;
- an exceptional capacity for original, creative and/or logical thinking;
- an excellent ability to organize, to analyze, to synthesize, to integrate ideas, and to express thoughts fluently.

Excellent (80-90) An excellent performance with strong evidence of

- a comprehensive grasp of the subject matter;
- an ability to make sound critical evaluation of the material given;
- a very good capacity for original, creative and/or logical thinking;
- an excellent ability to organize, to analyze, to synthesize, to integrate ideas, and to express thoughts fluently.

Good (70-79) A good performance with evidence of

- a substantial knowledge of the subject matter;
- a good understanding of the relevant issues and a good familiarity with the relevant literature and techniques;
- some capacity for original, creative and/or logical thinking;
- a good ability to organize, to analyze and to examine the subject material in a critical and constructive manner.

Satisfactory (60-69) A generally satisfactory and intellectually adequate performance with evidence of

- an acceptable basic grasp of the subject material;
- a fair understanding of the relevant issues;
- a general familiarity with the relevant literature and techniques;
- an ability to develop solutions to moderately difficult problems related to the subject material;
- a moderate ability to examine the material in a critical and analytical manner.

Minimal Pass (50-59) A barely acceptable performance with evidence of

- a familiarity with the subject material;
- some evidence that analytical skills have been developed;
- some understanding of relevant issues;
- some familiarity with the relevant literature and techniques;
- attempts to solve moderately difficult problems related to the subject material and to examine the

Course Overview

Principles and techniques for developing software combined with the practical experience of creating a mid-size software system as a member of a software development team. Includes: teamwork; projects, planning and process; users and requirements; use cases; modeling; quality; software architecture; testing; GUI design, design principles, patterns and implementation; ethics; professionalism.

The major workload in this course is a group project accomplished in self-selected teams of four people or less. Your team registers a company with the instructor with the goal of creating a software product for the course. It will be important to choose an appropriately scaled project that is neither too big or too small. The students are advised to talk to the instructor if they are not sure whether their project is of reasonable size. The tutorial leader and makers will also comment on the project size.

Discussions during the lectures will help establish your project scope. The milestones and the final project submission are the responsibility of the entire group, but grades will be assigned individually based on project results and individual contributions. As a result, grades may vary between team members. A work plan outlining the tasks assigned to each team member will be a part of each milestone submission. Teamwork is also a part of the project, and a portion of your grade will be assigned based on how well your team works together. The milestones focus on key deliverables for that stage in the project, but in order to complete the project in a timely fashion it will be important to work on deliverables from later stages early in the term.

Class Schedule

Tentative and might vary a bit over the term but we will mostly follow this below:

Week	**Module**	**Readings**	**Evaluation Due Date**
Sep 1	Introduction	The course: goals, content, workload, and expectations The course project	
Sep 8	Software Process	The nature of software and software engineering, Software failures	
Sep 22	Software Projects	Software process models, software projects, teams and planning, problem definition and problem scope	
Sep 29	Analysis	Basic cost estimation, Requirements and use cases, Business and domain modelling	

Oct 6	Design	Model based development, Introduction to design patterns, design principles	
Oct 13	Design	Mapping designs to implementation, Design principles and patterns	
Oct 20	Design	Advanced design patterns	
Oct 27	Testing	Quality assurance, Basic testing methods	
Nov 3	Human Computer Interaction	Usability engineering, GUI development, Quality assurance, Basic testing methods	
Nov 10	**Reading week, no lectures**		
Nov 17	Maintenance, Software architecture	The importance of maintenance Refactoring and re-engineering, Software architecture	
Nov 24	Project presentation and etc.	Student project presentations, Professionalism and Ethics	
Dec 1	Project presentations and exam overview	Exam overview, providing solutions on difficult exam related issues and so on, student project presentations	
	FINAL EXAM		

Midterm and Final Examination Scheduling

Midterm and final examinations must be written on the date scheduled.

Final examinations may be scheduled at any time during the examination period (INSERT FIRST AND LAST DAY OF CURRENT EXAM PERIOD); students should therefore avoid making prior travel, employment, or other commitments for this period. If a student is unable to write an exam through no fault of his or her own for medical or other valid reasons, documentation must be provided and an opportunity to write the missed exam may be given. Students are encouraged to review all examination policies and procedures:

http://students.usask.ca/academics/exams.php

Instructor Information

Contact Information

Chanchal Roy, Thorvaldson 280.4, chanchal.roy@usask.ca

Office Hours

There are no office hours. However, I am available anytime you ask for. Please email me if you have any questions or to arrange a time to meet. I give highest priority to meet my students and to talk about the concerns they have.

Instructor Profile

Chanchal Roy is an associate professor of Software Engineering/Computer Science at the University of Saskatchewan, Canada. While he has been working on a broad range of topics in Computer Science, his chief research interest is Software Engineering. In particular, he is interested in software maintenance and evolution, including clone detection, analysis and management, reverse engineering, empirical software engineering, and mining software repositories. He served or has been serving in the program committee of major software engineering conferences (e.g., ICSM, WCRE, MSR, ICPC and SCAM). He served as the Finance Chair for ICPC'11, Tool Co-chairs for ICSM'12 and WCRE'12, Tool Chair for SCAM'12, Poster Co-chair for ICPC'12, Program Co-chair for IWSC'12, Finance Chair for ICSM'13 and General Chair for ICPC'14. He has been serving as the General Chair for IWSC'15 and a Guest Editor of the Journal of Software: Evolution and Process. He has been teaching CMPT 370 for the last five years. Has been doing extensive research in Software Engineering in general, the core parts of this course. His teaching philosophy is provided below (in his words):

One of the main benefits of continuing my career as an academic is the opportunity to interact with, educate, and guide the next generation of students, both in a research laboratory setting as a mentor, and in a classroom setting as an instructor.

From my teaching and tutoring experience, I have found teaching to be tremendously rewarding, fascinating and enjoyable. Teaching provides me the opportunity to share the knowledge that I acquire through my education, research, and industrial and academic careers. Classroom teaching allows me to prepare students for their future careers. It also allows me to reevaluate and broaden my own understanding of course materials, not only through my lecture preparations and delivery, but also from the unique perspectives of individual students. There is also a strong connection between classroom teaching and research supervision. A good researcher becomes a valuable resource to those students in the classroom who are motivated to pursue their own research careers.

However, as a classroom teacher my aim is not just to educate and prepare, but also to motivate and inspire. I try hard to convey enthusiasm for the course topics. While my passion for teaching is the primary source of my enthusiasm, it also comes from recognition of the technical and societal implications of computer science topics. I try to maintain an active and discursive lecture style that increases student enthusiasm. The use of examples from real life, associating topics from the classroom with either current events (e.g., active research) or my professional experiences, helps show the students why the topics being covered are useful and important.

As a teacher I guide the students in the learning process. I noticed that most students are often unaware of their excellent intuitive grasps. I feel proud to see how motivated such students become when they can apply their intuition to the problems in the subject. My focus therefore is to create an environment where students feel free to explore the subject matter from various angles to develop ideas, definitions, models, applications and case studies. In particular, I always challenge the students to excel, foster collaborative discussion and group projects, and prompt individual critical thinking. It is also important that the students feel comfortable asking questions and providing feedback, as it helps them learn to articulate their thoughts and approaches, and in the same time, it helps me ascertain the changes I can incorporate to convey the course materials effectively.

Gaining students' trust is very important to me. I do this by being well organized in the management of the course materials, preparing relevant and stimulating assignments and practice problems to aid understanding of the curriculum, setting realistic goals, assigning reasonable course workload, being available outside the classroom, maintaining consistency in grading where hard work is repaid with either a good grade or specific reasons for failure and an opportunity for redemption, identifying and addressing misunderstanding early, and establishing close rapport between myself and the students and between the students themselves. In this way, we build a strong circle of trust and my students do well in the course on their own right.

Required Resources

None.

Readings/Textbooks

Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition, Craig Larman, Prentice Hall, 2005.

Object-Oriented Software Engineering: An Agile Methodology, David C. Kung, McGraw-Hill, 2013

Textbooks are available from the University of Saskatchewan Bookstore: www.usask.ca/consumer_services/bookstore/textbooks

Other Required Materials

None.

Electronic Resources

Will be available in Dropbox/Moodle.

Downloads

Supplementary Resources

Grading Scheme

Mid-term	15%
Assignments and Projects	35%
Final Exam	50%
Total	**100%**

Evaluation Components

Milestone No	Milestone Description	Mark	*Due Date
Milestone 0	A 0.5 (half) bonus mark for handling your project ideas (whatever you think) to us	**0.5**	Sep 14
Milestone 1	Form a group and send me the names and NSIDs of the group members (along with a group name)	**0**	Sep 17
Milestone 2	Early Project Proposal	**1.5%**	Sep 20
Milestone 3	Final Project Proposal with a toy prototype	**1.5%**	Sep 30
Presentation	Project proposal presentation	**2%**	TBA
Milestone 4	Requirements and preliminary design	**6%**	Oct 15
Milestone 5	Detailed design of work units	**6%**	Nov 3
Presentation	Project presentation	**3%**	TBA
Milestone 6	**Final project**	**15%**	Nov 27

*****The due dates are kind of tentative. A firm due date will be provided when the assignment is posted.

Midterm Exam

Value: 15% of final grade
Date: Nov 3, 2014
Length: 40 minutes
Type: Closed book
Description: Will cover the materials until the Oct 31, 2014

Final Exam

Value: 50% of final grade
Date: See Course Schedule
Length: 3 hours
Type: Close book
Description: Will cover all the materials covered in the lectures.

Submitting Assignments

Via Moodle.

Late Assignments

Negotiable depending on the nature of the students' problems.

Criteria That Must Be Met to Pass

The students have to pass both the final exam. Furthermore, the students should at least do reasonably in the course projects with at least passing grade.

Attendance Expectations

While attendance is not mandatory, the students are recommended to attend the lectures. There will be a couple of bonus marks for the students who attend regularly.

Participation

Participating in the group discussion either in the group project and/or within the lectures are highly recommended and there might be a couple of bonus marks on this.

Student Feedback

Students' feedback both during the lectures and at the end of the term evaluations will be seriously considered to improve the course offering.

Integrity Defined (from the Office of the University Secretary)

The University of Saskatchewan is committed to the highest standards of academic integrity and honesty. Students are expected to be familiar with these standards regarding academic honesty and to uphold the policies of the University in this respect. Students are particularly urged to familiarize themselves with the provisions of the Student Conduct & Appeals section of the University Secretary Website and avoid any behavior that could potentially result in suspicions of cheating, plagiarism, misrepresentation of facts and/or participation in an offence. Academic dishonesty is a serious offence and can result in suspension or expulsion from the University.

All students should read and be familiar with the Regulations on Academic Student Misconduct (http://www.usask.ca/secretariat/student-conduct-appeals/StudentAcademicMisconduct.pdf) as well as the Standard of Student Conduct in Non-Academic Matters and Procedures for Resolution of Complaints and Appeals (http://www.usask.ca/secretariat/student-conduct-appeals/StudentNon-AcademicMisconduct.pdf)

For more information on what academic integrity means for students see the Student Conduct & Appeals section of the University Secretary Website at:
http://www.usask.ca/secretariat/student-conduct-appeals/forms/IntegrityDefined.pdf

Examinations with Disability Services for Students (DSS)

Students who have disabilities (learning, medical, physical, or mental health) are strongly encouraged to register with Disability Services for Students (DSS) if they have not already done so. Students who suspect they may have disabilities should contact DSS for advice and referrals. In order to access DSS programs and supports, students must follow DSS policy and procedures. For more information, check http://www.students.usask.ca/disability/, or contact DSS at 966-7273 or dss@usask.ca.

Students registered with DSS may request alternative arrangements for mid-term and final examinations.
Students must arrange such accommodations through DSS by the stated deadlines. Instructors shall provide the examinations for students who are being accommodated by the deadlines established by DSS.

Acknowledgements

Course Contributor(s)