

COSC 310 – Software Engineering

Winter 2006

Instructor: Dr. Ramon Lawrence
Class Schedule: 8:30 a.m. – 9:30 a.m. Monday/Wednesday/Friday
Location: SCI 236
Lab time/location: 2:00 p.m. – 4:00 p.m. Wednesdays at SCI 234
Office Hours: 1:00–3:00 p.m. Monday and 2:00-4:00 p.m. Thursday or by appointment
Office Location: SCI 263
Phone: 807-9390
E-mail: ramon.lawrence@ubc.ca (preferred contact method)
Course URL: <http://people.ok.ubc.ca/rlawrenc/teaching/310/index.html>

Course Description

Official Calendar: Design and implementation of large, multi-module program systems. Software life cycle. Design tools. Features and use of module-orientated programming languages. Intermodule communication. OUC equivalent: COSC 310.

Specific description: This course provides an introduction to software engineering including the fundamental activities of specification, design, implementation, testing, and maintenance. Students learn about various life-cycle models for organizing these activities, and the importance of a software process to create quality software. The laboratory sessions cover software development tools, UML modeling techniques, and teach good development practices. A major component of the course is a semester long, group project where students apply the software engineering fundamentals to a medium-scale development project.

Prerequisite

- COSC 222 – Computer Data Structures
- Co-requisite: COSC 304 – Introduction to Database Management Systems

Marking and Evaluation

Assignments	20 %
Project	40 %
Midterm Exam	15 % (50 minutes in class)
Final Exam	25 % (cumulative)

No late assignments will be accepted.

Textbook and Reference Material

Recommended: Ian Sommerville, *Software Engineering* 8, Addison -Wesley, 8th edition, ISBN 978-0-321-31379-9, 2007.

Expectations

- I expect students to attend **all** classes and prepare before attending class. This includes reading relevant sections of the textbook and reviewing notes from previous lectures.
- I recommend all students download and read a copy of the lecture notes **before** the lecture.
- I expect all students to learn the material in the course and undertake sufficient effort to produce all the programming assignments and quality projects.
- I want all students to enjoy attending class and feel free to participate according to their own personalities. Feel free to ask questions by raising your hand or speaking out at appropriate times.
- Please actively participate in class discussions, questions, and problem solving exercises.
- **I want all students to pass the course, receive a good grade, and feel the course was beneficial.**

Homework Expectation

For this course, it is expected that you will spend *at least six hours per week in out-of-class preparation.*

Grievances and Complaints Procedures

A student who has a complaint related to this course should follow the procedures summarized below.

- The student should attempt to resolve the matter with the instructor first. Students may talk first to someone other than the instructor if they do not feel, for whatever reason, that they can directly approach the instructor.
- If the complaint is not resolved to the student's satisfaction, the student should go to the departmental chair Cynthia Mathieson, at ART 300, 807-8730.

Your Responsibilities

Your responsibilities to this class and to your education as a whole include attendance and participation. You have a responsibility to help create a classroom environment where all may learn. At the most basic level, this means you will respect the other members of the class and the instructor and treat them with the courtesy you hope to receive in return. Inappropriate classroom behavior may include: disruption of the classroom atmosphere, engaging in non-class activities, talking on a cell-phone, inappropriate use of profanity in classroom discussion, use of abusive or disrespectful language toward the instructor, a student in the class, or about other individuals or groups.

Academic Dishonesty

The academic enterprise is founded on honesty, civility, and integrity. As members of this enterprise, all students are expected to know, understand, and follow the codes of conduct regarding academic integrity. At the most basic level, this means submitting only original work done by you and acknowledging all sources of information or ideas and attributing them to others as required. This also means you should not cheat, copy, or mislead others about what is your work. Violations of academic integrity (i.e., misconduct) lead to the break down of the academic enterprise, and therefore serious consequences arise and harsh sanctions are imposed. For example, incidences of plagiarism or cheating usually result in a failing grade or mark of zero on the assignment or in the course. Careful records are kept in order to monitor and prevent recidivism. A more detailed description of academic integrity, including the policies and procedures, may be found at <http://web.ubc.ca/okanagan/faculties/resources/academicintegrity.html>, and in the calendar at <http://okanagan.students.ubc.ca/calendar/index.cfm?tree=3,54,111,0>. If you have any questions about how academic integrity applies to this course, please consult with your professor.

Students with Disabilities

If you require disability related accommodations to meet the course objectives please contact the Coordinator of Disability Resources located in the Student development and Advising area of the student services building. For more information about Disability Resources or about academic accommodations visit <http://okanagan.students.ubc.ca/current/disres.cfm>.

Missing an Exam

Only students who miss an exam for a reason that corresponds to the University of British Columbia Okanagan's policy on excused absences from examinations will be permitted to take the exam at a later time. Please note that a make-up exam may have a question format that is different from the regular exam. If the reason for absence is satisfactory, the student may either take the exam, or if a midterm exam is missed, the midterm exam can be forfeited and the student's final exam will be worth more of the final grade.

Course Outline

The course has a substantial amount of material to be covered in a short time. This requires the student make a strong effort to keep up with the material discussed in class. Below is an outline of the topics covered. The professor is not bound to the topics, timelines, and outline provided as they only serve as a general reference.

Laboratory times: During the laboratory time we will discuss assignment questions and go over past assignments. Each laboratory time will have a defined topic with the majority of labs involving use of software engineering software and project development practice.

Date	Topics Covered and Description
September 6 (W)	Introduction to course and course goals. Project overview. Lab 1: Student skills assessment. Select project topic.
September 8 (F)	Overview of Software Engineering – motivation, benefits, profession (Ch. 1)
September 11 (M)	Software Process – The fundamental software development activities (4.1,4.3)
September 13 (W)	Software Process Models – Waterfall, evolutionary models, spiral models (4.1,4.2,4.4) Lab 2: Improving my own software process
September 15 (F)	Software Process Models (cont.) – extreme programming, model comparison (Ch. 17)
September 18 (M)	Requirements Specification phase (Chapters 6 & 7)
September 20 (W)	Requirements Specification phase (cont.) - use-case modeling Lab 3: Requirements gathering practice
September 22 (F)	Project requirements meeting and discussion
September 25 (M)	Requirements Specification phase (cont.) – requirements document
September 27 (W)	Design phase – data flow diagrams (Chapters 11, 12, 13) Lab 4: Building data flow diagrams
September 29 (F)	Design phase (cont.) – architecture decisions
October 2 (M)	Object-oriented design – Class modeling using UML Diagrams (8.4,14.1)
October 4 (W)	Object-oriented design – Class modeling using UML Diagrams (cont.) Lab 5: UML class diagrams using JUDE
October 6 (F)	Project design meeting
October 9 (M)	No class for Thanksgiving.
October 11 (W)	Object-oriented design – Dynamic UML Diagrams Lab 6: UML sequence diagrams using JUDE
October 13 (F)	Project user interface design meeting
October 16 (M)	Object-oriented design – Overall Design Process (14.2)
October 18 (W)	Midterm review. Lab 7: Midterm review.
October 20 (F)	Midterm exam in class.
October 23 (M)	Project design/implementation meeting
October 25 (W)	User Interface Design (Chapter 16) Lab 8: Pair-programming - Is it better?
October 27 (F)	Implementation Phase: Software development, extreme programming (Chapter 17)
October 30 (M)	Software re-use (Chapter 18)
November 1 (W)	Verification and Validation (Chapter 22 and 23) Lab 9: Automated testing tools and software inspections
November 3 (F)	Project implementation feedback meeting #1
November 6 (M)	Maintenance: Software evolution (Chapter 21)
November 8 (W)	Configuration Management (Chapter 29) Lab 10: The Maintenance Challenge and Introduction to CVS
November 10 (F)	Project Management – People Management (Chapter 25)
November 13 (M)	No class for Remembrance Day.

November 15 (W)	Project Management – Time Management and Planning (Chapter 5) <i>Lab 11: Introduction to Microsoft Project</i>
November 17 (F)	Project implementation feedback meeting #2
November 20 (M)	Quality Management (Chapter 27)
November 22 (W)	Project Cost Estimation – Overview (Chapter 26) <i>Lab 12: Software costs: How long and how much?</i>
November 24 (F)	Final Project Meeting
November 27 (M)	Software Process Improvement – ISO 9000 Certification (Chapter 28)
November 29 (W)	Final project presentations <i>Lab 13: Final exam review</i>
December 1 (F)	Last day of class. Class evaluations and review for final exam.