UNIVERSITÉ Concordia UNIVERSITY

CONCORDIA INSTITUTE FOR
Information Systems Engineering

Concordia University

Concordia Institute for Information Systems Engineering

INSE 6260/4/UU - Software Quality Assurance (Winter 2009)

Time:	Thursday, 17h45 – 20h15
Classroom:	H-423
Office hours:	Wednesday, 14h00 – 16h00 or by appointment (Office: EV.7.630)

Instructor:
Dr. J. Bentahar
E-mail: bentahar@ciise.concordia.ca
Tel: 514-848-2424 Ext. 5382

Course Web:
http://www.ciise.concordia.ca/~bentahar/inse6260.html
You can use this web site to get lecture notes, useful links, assignments, and other useful information. It is highly advised to visit the web site regularly.

Textbooks:

Suggested Readings

1) *Software Quality Assurance: From Theory to Implementation*
Daniel Galin
Addison Wesley, 2004
ISBN: 0-201-70945-7
This book covers several issues related to software quality assurance. Some important chapters are: software quality challenge, what is software quality, software quality factors, and software testing.

Suggested Readings

2) *Metrics and Models in Software Quality Engineering*
Stephen H. Kan
Addison-Wesley, 2004 (available online)
ISBN: 0-201-72915-6
This book is a reference in software metrics. It covers a comprehensive breadth of measurement theory and software quality metrics.

Suggested Readings

3) *Foundation of Testing*
Aditya P. Mathur
Draft book

Description:

This course presents the main issues of quality assurance for software engineering. It introduces software quality challenges and factors and the main quality considerations for software. The following issues will be covered: quality assurance, quality factors, components of a software quality assurance system, contract review, software development and quality plans, activities and alternatives, integration of quality activities in a project lifecycle, reviews, software inspection, software verification, testing processes, static analysis, control-flow analysis, data-flow analysis, control-flow testing, loop testing, data-flow testing, transaction-flow testing, domain testing, type-based analysis, dynamic analysis, usage models, operational profiles, result and defect analysis, reliability, performance analysis, maintenance and reverse engineering, case tools and software quality assurance. Students will discover various concepts and techniques developed in recent research about software quality engineering and learn to apply them through lectures, readings, assignment, and team project. Several materials from different sources will be used, particularly scientific papers.

Prerequisites:

INSE 6210, COMP 5541 or equivalent.
Good knowledge of mathematics and logics will be very helpful.

Requirements:

- One individual/group assignment
- One in-class midterm exam (closed book)
- One in-class final exam (closed book)
- One team project (2~3 members, presentation + report)

Grading:

- One assignment: 15%
- One in-class midterm exam: 25%
- One in-class final exam: 30%
- Project (presentation + report): 15% + 15% = 30%

The assignment will be graded based on **clarity** and **comprehensiveness**. The project will be graded based on **originality**, **relevance of the proposed solution**, and **contribution**. In-class exams will test students' knowledge and ability to understand, analyze, and synthesize concepts.

Important dates:

- Project proposal: February 05, 2009
- Assignment: February 12, 2009
- Midterm exam: March 05, 2009
- Project presentation: April 02, 2009
- Project report: April 10, 2009
- Final exam: To be scheduled

Useful Links:

The Software Quality Page
http://www.swquality.com/users/pustaver/index.shtml

Software QA and Testing Resource Center
http://www.softwareqatest.com/

Formal Methods
http://vl.fmnet.info/

Software Quality Institute (SQI)
http://lifelong.engr.utexas.edu/sqi/index.cfm

Submission:

All assignments are at the beginning of class. Late assignments will incur a penalty of 20% deduction (up to 100%). No points will be given to the assignment submitted 5 days after the due date.

Policies:

Cheating and plagiarisms will be very seriously considered and handled according to the Concordia Academic Code of Conduct (can be found in the graduate student handbook) without exception. Please note the schedule of the exams. A makeup test will be given only in the case of a serious illness or emergency. You must contact the instructor before the exam. Only written and proved documentations are accepted for verification purposes.

Tentative Schedule: The table below provides a brief summary of some of the material that will be covered during the term. The schedule may change slightly.

Lecture	Topic	Event
1	Introduction to Software Quality Assurance	
2	Software Quality Factors, Models and Standards	
3	Inspection: Verification and Validation	
4	Introduction to Testing	
5	Test Assessment	Project Proposal **(February 5)**
6	Testing Techniques	Assignment **(February 12)**
7	Reachability Analysis	
8	Break	
9	Midterm	Midterm **(March 5)**
10	Structural and Mutation Testing	
11	SDL	
12	Software Metrics	
13	Project Presentations	**(April 2)**
14	Project Presentations + Final Exam	