# Course Outline

| Course Name: | Software Engineering Methods  
COMP-335 Winter 2010 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor:</td>
<td>Joseph Vybihal</td>
</tr>
</tbody>
</table>
| Contact Information: | Office: ENGMC 323  
Office Hours: Tue 2-3PM MC323, Fri 1-2PM TR labs (or by appointment)  
(in Trottier 3rd floor labs - look for me)  
Email: Use the WebCT email tool  
Emergencies or non-course related: jvybihal@cs.mcgill.ca |
| Course Objectives: | (1) To learn how to develop software in a team environment.  
(2) To learn how to write a requirement, specification and design document.  
(3) To learn how to work on real software projects.  
(4) Get exposure to different software engineering case studies.  
(5) Learning and using UML in real software engineering situations. |
| Course Description: | This course is designed to introduce the computer science student to the essentials of software engineering. Often programmers do not place much thought into formulating a structured methodology for designing and constructing large scale programming project. This course attempts to build the foundation for such a framework. Important techniques need to be developed in problem analysis, software design (thinking), documentation and programming. Explorations in creating software, communicating with clients, creating a requirement document, planning a project and working in a team will be developed. Being a good programmer is not only being able to write code that compiles and runs well. **A good software engineer combines science with business and art.** It is a complex combination of conflicting priorities that must be brought into synergy – a maximization of competing issues. Only then, when code looks pretty and runs efficient while satisfying the needs of the client and communicated in such a way that everyone knows what to do, only then do we have a good project. |
| Texts: | Primary Text:  
| Supplementary Texts: | • Essentials of Software Engineering, by Frank Tsui & Orlando Karam  
ISBN: 978-0-7637-3537-1 (Jones and Bartlet)  
• Software Architecture and Design Illuminated, Qian/Fu/Tao/Xu & Diaz, Jones and Bartlet, ISBN 978-0-7637-5420-4  
• Use Cases, Kulak & Guiney, Addison Wesley, ISBN 0-201-65767-8 |
| Evaluation: | • 3 Assignments worth 10%  
• 1 Group project (2-4 in a team) worth 30% each  
• Midterm exam worth 10% (March 3, 2010 – in class exam)  
• Final exam worth 50% |
You will be notified in advance of assignment due dates. All assignments are due on WebCT at the indicated time and date. Late assignments will lose 5% of its grade per day late. Assignments beyond 2 days late will not be accepted. You may not submit assignments via e-mail without the permission of the instructor.

Students are responsible for all materials for the tests and exams, whether or not it is covered in class. Exams will be a combination of all types of questions based on all sources, and students may be required to integrate theoretical concepts from the text to substantiate their arguments.

No make-up tests or make-up assignments are allowed in this course. A supplemental exam is possible for 50% of the grade (to replace your final exam).

If you are not satisfied with the grading of an assignment or mid-term test, you may request a review within 7 days of return. Indicate in writing or during a meeting with the instructor where and why you feel the marks are unjustified and give it back to your instructor for re-grading. Note that the entire assignment or mid-term test will be re-graded and your grade can go up or down (or stay the same) accordingly.

**Course Requirements:** Students will already have taken a programming course like COMP-302 before participating in this course.

**Calculators**
Only non-programmable, no-tape, noiseless calculators are permitted. Calculators capable of storing text are not permitted in tests and examinations.

**Dictionaries**
Dictionaries are not permitted, but translation dictionaries are.

**Handheld Devices**
Handheld devices capable of storing text and having calculator functionality (e.g. Palm, etc.) are not permitted.

**Additional Information:** The course slides are not meant as a complete set of notes or a substitute for a textbook, but simply constitute the focus of the lecture. Important gaps are left in the slides that are filled in during class, thus lecture attendance should be considered essential.

The material covered in the classroom will be used to supplement textbook readings. Every chapter should be read twice. The first reading should be done prior to attending class and the second reading should be done after the class discussion of the chapter. The questions at the back of each chapter follow directly from the reading. Students should be able to answer these questions after a thorough reading of the material.

**Academic Integrity:** *Code of Student Conduct*
McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see [www.mcgill.ca/integrity](http://www.mcgill.ca/integrity) for more information).

L'université McGill attache une haute importance à l'honnêteté académique. Il incombe par conséquent à tous les étudiants de comprendre ce que l'on entend par
tricherie, plagiat et autres infractions académiques, ainsi que les conséquences que peuvent avoir de telles actions, selon le Code de conduite de l'étudiant et des procédures disciplinaires (pour de plus amples reseignements, veuillez consulter le site www.mcgill.ca/integrity).

Final Exam Policy:  

Regulations  
Students should not make other commitments during the final exam period. Vacation plans do not constitute valid grounds for the deferral or the rescheduling of examinations. See the Centre Calendar for the regulations governing Examinations:  
http://www.mcgill.ca/conted-students/exams/regulations/  
Students are required to present their I.D. Card (with photo) for entrance to their examination.

Conflicts  
If you are unable to write your final examination due to scheduling conflicts, you must submit a Final Exam Conflict Form with supporting documentation at least one month before the start of the final examination period. Late submissions will not be accepted. For details, see  
http://www.mcgill.ca/conted-students/exams/conflicts/

Exam Timetable  
Examination schedules are posted at the Centre and on the following page approximately 6-8 weeks before the examination period commences  
The Centre cannot provide examination dates over the telephone.

Classroom Rules:  
All electronic devices (cell phones and beepers) must be turned off during class time.

Assignments Pickup:  
All assignments are submitted to and picked-up from WebCT.

Email Policy:  
E-mail is one of the official means of communication between McGill University and its students. As with all official University communications, it is the student's responsibility to ensure that time-critical e-mail is accessed, read, and acted upon in a timely fashion. If a student chooses to forward University e-mail to another e-mail mailbox, it is that student's responsibility to ensure that the alternate account is viable. Please note that to protect the privacy of the students, the University will only reply to the students on their McGill e-mail account.

Computing Resources:  
Trottier 3rd floor.

Students Rights and Responsibilities:  
Regulations and policies governing students at McGill University can be downloaded from the website:  
http://www.mcgill.ca/deanofstudents/rights/  

Students Services and Resources:  
Various services and resources, such as email access, walksafe, library access, etc., are available to Continuing Education students:  
http://www.mcgill.ca/conted-students/services/  

Minerva for Students:  
http://www.mcgill.ca/minerva-students/
<table>
<thead>
<tr>
<th>WEEK</th>
<th>TOPIC</th>
<th>DETAILS</th>
<th>CHAPTER</th>
<th>WORK</th>
</tr>
</thead>
</table>
| 1    | Developing the Plan | A) The instability of software & correctness  
B) Making a plan  
C) Control | 1 & 2 |  |
|      | Elements of a Plan | (Intro. to Soft. Engineering) | | |
| 2    | Software Process and Process Models | A) Software Life cycles  
Classical, Modern, Domain  
B) Process Models  
C) Process Assessment  
D) Choosing a process | 4 & Notes | Ass 1: Find Client & Life cycles |
| 3    | Dependable Systems | A) Standards docs, methods, IEEE, ISO  
B) “A Good Program” and Quality models  
C) Clean Room Programming  
D) Risk Management | 3, 27, 28 & Notes |  |
| 4    | Software and Team Development + Management | A) The Project Environment stakeholders, team organization  
B) Agile Processes  
C) Keeping client in the loop | 5, 17, 25 | Ass 2: Team & Software |
|      | The Project Environment | | | |
| 5    | Engineering the program | A) Good Programming Practices  
(C, modules/objects, reusability)  
B) Team Programming Practices  
C) The Peer Review Technique  
D) Code Management Tools | 18, 20, 29 & Notes |  |
| 6    | Requirements and Specifications | A) How to gather requirements  
B) Effort Estimation  
C) The Preliminary Design | 6, 7, 26 & Notes | Project Part 1: Preliminary Design |
| 7    | Requirements Engineering | | | |
| 8    | Review | | 9, 10, Notes | Midterm |
| 9, 10, 11 | Problem Decomposition and Design | | | |
|      | Real-life Case Studies | A) Types of programs  
UI, Event, Real-time based  
B) Types of models  
Static vs. Dynamic Models:  
Weighted Casual Graph  
State-Transition Graph  
Design patterns  
C) Representation techniques  
UML & Use Case & OO | 11, 12, 13, 14, 15, 16, Notes | Project Part 2: Req. Design Document and Software |
| 12   | Design & Specifications | Design Characteristics  
Design Metrics  
**Meta-Architecture:** Data flow, Data-Centered, Hierarchical, Asynchronous Comm, Interaction-Oriented, Distributed, Component-based, Heterogeneous, UI, & Product | 8 & Notes | Ass 3: Design & Specs + Misc. |
| 13   | Quality Control and Testing | A) Client’s Concerns  
B) Unit Testing  
C) Refactoring & Regression Testing  
D) Migration, archiving & installing | 21, 22, 23, 24 |  |