# Carleton University

## Department of Systems and Computer Engineering

**SYSC-4800 Software Engineering Fall 2011**

## Course Outline

**Instructor:** Yvan Labiche 520-2600 ext. 5583
Room 7078 MC labiche@sce.carleton.ca
Office hours: Tuesdays and Thursdays from 13h to 14h.
Otherwise, by appointment.

**Course Description and Objectives:**

Software engineering is concerned with the theories, methods, and tools that are needed to develop complex, large-scale software. It encompasses a wide range of topics, including requirements and specification, software design, implementation, validation and verification, software maintenance, and the management of the software process. Every single software development process involves one or more of these topics. It is therefore paramount to understand them all. With the Unified Modeling Language (UML) becoming the industry, de-facto standard notation for software development (70% of IT industry uses UML), software development is becoming more and more model-driven (or model-based), with less manual generation of source code but more automated generation of source code (from models).

A single course is clearly incapable of covering all these topics in depth.

The goal of this course is to provide you with a broad understanding of all phases in the model-driven, software life-cycle, and to introduce you to specific concepts that you have not covered systematically yet, and that are increasingly being viewed as essential for engineering large software systems.

More specifically, the five main goals of this course are:

- To understand the role of each phase of a software development life cycle and to understand how, together, they make up the entire life cycle;
- To understand the main principles that underlie these phases;
- To master model-based software development by using the UML;
- To understand the challenges of software evolution;
- To understand the challenges of software verification and validation.

And the three main objectives of this course are:

- To conduct requirement elicitation, thereby producing software requirements under the form of a use case model, using established principles;
- To produce an analysis models made of consistent diagrams (class, sequence, state machine diagrams), following well established heuristics;
- To conduct system design and object design by using patterns.

**Prerequisites:**

SYSC-3001, SYSC-3100 and SYSC-3303 are the prerequisite for SYSC-4800:

- SYSC-3001 provides some background material necessary during system design (part of the software life cycle);
- SYCS-3100 provides some background on the UML notation: I expect students to "speak" the UML fluently;
- SYSC-3303 provides some background material necessary during system design;

Students who have not satisfied the prerequisites for this course must either a) withdraw from the course, or b) obtain a prerequisite waiver from www.sce.carleton.ca/ughelp, or c) may be deregistered from the course after the last day to register for courses.

## Course Outline

**Lectures:**

When: Wednesdays and Fridays from 11h30 to 13h.
Where: TBD

**Laboratory Sessions (compulsory):**

When: Mondays (every other week, starting Sept. 12) from 11h30 to 14h30.
Except Oct. 10th.
Where: TBD

**Textbook:**

I will use the following textbook as a primary reference:

Bernd Bruegge and Allen Dutoit, *Object-Oriented Software Engineering: Using UML, Patterns and Java*, Third Edition, Prentice-Hall 2009, ISBN 0-13-606125-7.

Although course material will draw extensively from other sources—there are quite a few useful Software Engineering books that each present their own strengths and these are listed under "recommended readings" on the website—this textbook provides an excellent basis for this course. Students are therefore encouraged to purchase it.
I will also make available the course transparencies in a PowerPoint format on the course web site (WebCT), on a weekly basis.

**Grading Scheme:**

There will be:

- three assignments worth 15% of the final mark, i.e., each assignment is worth 5% of the final mark;
- a (closed-book) midterm exam weighting 25%;
- a (closed-book) final exam worth 30%;
- compulsory lab work worth 30%: each of the 6 labs is worth 5% of the final mark.

**To pass the course,** an appropriate overall mark (D- or higher) must be obtained and a passing mark (D- or higher) must be obtained at the final exam.

**Important dates:**

- Assignment 1 will be posted on Sept. 29 and due on Oct. 27.
- Assignment 2 will be posted on Oct. 28 and due on Nov. 17.
- Assignment 3 will be posted on Nov. 18 and due on Dec. 2.
- The (in-class) midterm is scheduled on Oct. 12.
- The final exam will be scheduled during the December exam period.

Students with conflicts: see Academic Accommodation below.

**Assignments:**

The assignments will lead you through the software development steps, and are a good starting point when preparing the mid-term and final exams. Portions of the work from each assignment may be reused and refined in subsequent assignments, so students are encouraged not to "write-off" any particular assignment just because of its relatively low weight in the overall grading scheme.
Students are encouraged to discuss issues when working on assignments; however, you are expected to submit your own work for grading. There is a fine line between cooperating with your colleagues (discussing problems and ideas) and copying solutions (plagiarism). Not only is

plagiarism an instructional offence (see the Undergraduate Calendar), but doing the assigned work by yourself is by far the best way to prepare for the exams.
Submission: Assignments are due at midnight of the due date and must be submitted on WebCT.
Late assignments will be graded according to the following policy: a 20% penalty per day (i.e., 24 hours).

**Final Exam:** *Is for evaluation purposes only and will not be returned to the student.*
Students who miss the final exam may be granted permission to write a deferred examination (see the Undergraduate Calendar for regulations on deferred exams).

**Academic Accommodation**
You may need special arrangements to meet your academic obligations during the term. For an accommodation request the processes are as follows.
Pregnancy obligation: write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details visit the Equity Services website
http://www.carleton.ca/equity/accommodation/student_guide.htm
Religious obligation: write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details visit the Equity Services website
http://www.carleton.ca/equity/accommodation/student_guide.htm
Students with disabilities requiring academic accommodations in this course must register with the Paul Menton Centre for Students with Disabilities for a formal evaluation of disability-related needs. Registered PMC students are required to contact the Centre, 613-520-6608, every term to ensure that I receive your Letter of Accommodation, no later than two weeks before the first assignment is due or the first in-class test/midterm requiring accommodations. If you require accommodation for your formally scheduled exam(s) in this course, please submit your request for accommodation to PMC by November 11th 2011 for Fall term (December exams).
You can visit the Equity Services website to view the policies and to obtain more detailed information on academic accommodation at http://carleton.ca/equity/accommodation

**Plagiarism:**
Plagiarism (copying and handing in for credit someone else's work) is a serious instructional offense that will not be tolerated. Please refer to the section on instructional offenses in the Undergraduate Calendar for additional information.

**Health and safety:**
http://www.sce.carleton.ca/courses/health-and-safety.pdf

## Course Outline

**Tentative Outline**

The following is a tentative outline of the course; it might change, based on time constraints:

| | |
|---|---|
| Week 1 | Introduction to Software Engineering<br>the nature of software, history and scope of software engineering, relationships with other fields, fundamental principles, software life cycle |
| Week 2 | Requirement Elicitation using UML<br>Producing a specification of the system that the client understands. Relationship between requirements and specifications, the uses of specifications, the qualities of specifications, the requirements engineering process and products. |
| Week 3 | |
| Week 4 | Object-Oriented Analysis using UML<br>Producing an analysis model that the developers can unambiguously interpret. Formalizing the requirements (requirement elicitation) into specifications (Analysis). |
| Week 5 | |
| Week 6 | System Design using UML<br>Definition and objectives, object-oriented design with UML, architectural design, detailed design (with strong emphasis on design patterns), concurrent software, safety analysis and fault tolerance. |
| Week 7 | |
| Week 8 | |
| Week 9 | Revisiting Design Patterns |
| Week 10 | |
| Week 11 | Object Design<br>Optimizing software architecture, optimizing class diagram. |
| Week 12 | |
| Week 13 | Other Software Engineering Issues<br>Fault tolerance, real-time systems, verification and validation, re-engineering |

**Miscellaneous:**

Attendance:

Being (consistently) late in lectures and/or labs does not show professional behavior. Moreover, those students who miss lectures and/or labs may encounter difficulties as far as their final grade is concerned since (i) a lot of the material introduced during lectures is not necessarily in the transparencies provided on the course web site (this may depend on student participations, student questions, …), and (ii) questions during the final exam will target the understanding of lab activities as well as lab material.

Expectations:

I expect the students to invest a substantial amount of time and energy in reading the textbook and doing the assignments/lab work. Looking at the transparencies I will provide may not be enough to achieve the level of understanding that will be required for the mid-term and final exams.

**Let us try to avoid this…**