Course Outline

Professor: Dr. Dorina Petriu
Office hours: Thursdays, 4:00 - 5:00 pm
Room 5205 Canal Building
Tel: (613) 520-2600 x5652
Otherwise email me for an appointment.
Email: petriu (at) sce (dot) carleton (dot) ca

Calendar Description:
Introduction to software engineering principles, software development life-cycles. Modelling in software engineering. Current techniques, notations, methods, processes and tools used in software engineering. UML modelling. Introduction to software quality, software verification and validation, software testing.

Precludes additional credit for SYSC 3100, SYSC 3120, SYSC 4120 and COMP 3004.
Prerequisite(s): SYSC 2004 and (SYSC 2006 or SYSC 2002).

Course Description and Objectives:
Software engineering is concerned with the theories, methods, and tools for developing complex, large-scale software. It encompasses a wide range of topics, including requirements elicitation and specification, software design, software construction (i.e., implementation), validation and verification, software maintenance, and the management of the software process. Every software development project involves one or more of these activities. With the Unified Modeling Language (UML) becoming the de-facto standard notation for software development in the IT industry, software development is becoming increasingly model-driven (or model-based), with less manual generation of source code and more automated generation of source code (from models).

A single course is clearly incapable of covering all these topics in depth. The aim of this course is to provide you with a broad understanding of the phases and activities in model-driven software development, and to introduce you to specific concepts that have not been covered systematically in first-year and second-year programming courses, yet are widely regarded as essential for engineering large software systems.

More specifically, the goals of this course are:
1. To understand how the software development life cycle consists of multiple phases, to understand the role of each phase, the relationships between them, and the main principles that underlie these phases.
2. To learn model-based software development, using the UML to render the models.
3. To understand the challenges of software evolution.
4. To understand the challenges of software verification and validation.

The three main objectives/outcomes of this course are:
1. Students should be able to conduct requirement elicitation and produce software requirements in the form of use case models.
2. Students should be able to produce analysis models consisting of UML models (composed of class, sequence, state machine diagrams), following well-established heuristics.
3. Students should be able to conduct system design and object design using patterns.
Prerequisites
Students who have not satisfied the prerequisites for this course must: a) withdraw from the course; b) obtain a prerequisite waiver from www.sce.carleton.ca/ughelp, or c) may be deregistered from the course after the last registration deadline.

Lectures:
When: Wednesdays and Fridays, 4:00 - 5:30 pm
Where: ME 4499

Laboratory Sessions:
When: Mondays, 2:30 - 5:30 pm (alternate weeks)
Where: AA 508

Textbook:
The following textbook will be the primary reference:

Although the course material will draw extensively from other sources, this textbook provides an excellent basis for this course. Students are therefore encouraged to purchase it. The course slides will be made available in PDF format on the course web site, on a weekly basis.

Important notice: Course notes are subject to changes without notice until presented in class. After being discussed in class, changes to course notes will be posted online.

Marking Scheme:
- 20% Assignments (3)
- 10% Labs (6)
- 5% In-class unannounced Quizzes
- 20% Mid-Term Exam (TBD, in-class, closed-book)
- 45% Final Exam (Centrally scheduled, 3 hours, closed-book).

Important Notes

1. **To pass this course**, a student must obtain an appropriate overall mark (D- or higher), a passing mark (D- or higher) for the final exam and get credit for at least five out of six labs.

2. **The final exam** will be held during the formal examination period set out in the University Calendar and will be scheduled by Exam Services. The instructor will not accommodate any special requests or alternate arrangements. The 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).

3. **Students who miss the midterm** due to illness must provide a valid medical certificate to the instructor no later than 48 hours after returning to campus. The certificate must clearly state the name of the doctor with contact information. Once the certificate has been verified, a make-up midterm examination will be arranged (outside of the class hours).

4. **Students who have previously taken SYSC 3020**: please note that there are no lab exemptions from this course.
Lab Periods:

Attendance at the scheduled laboratory periods is mandatory, and attendance will be taken. During the labs you will work on short exercises that are intended to provide practical experience with tools and techniques related to the concepts presented in the lectures. You will normally be required to demonstrate and submit your lab work by the end of the lab period (or other specified deadline), as indicated in that week's lab handout.

Your work in each lab period will be graded satisfactory, marginal, or unsatisfactory.

- **Satisfactory** (75-100%) means that you were present at the lab and made reasonable progress towards completing the lab exercises. Note that you do not have to finish all the exercises to receive a satisfactory grade.

- **Marginal** (50-75%) means that you made some progress towards completing the exercises, but your solutions were not sufficiently complete to warrant a satisfactory grade. This grade indicates that you may be falling behind, and should take steps to remedy this situation.

- **Unsatisfactory** (0-50%) means that you were absent from the lab period, or you attended but made little or no progress towards completing the lab exercises. This indicates that you are likely having difficulty understanding important concepts and should seek help from your instructor as soon as possible. You will also receive unsatisfactory if you do not demonstrate or submit your work before the deadline or if it is apparent to the TA that you did not do enough of the lab work on your own; that is, you relied on your colleagues to explain the exercises and provide solutions. If you are absent from a lab period for any reason, it is up to you to do the missed lab work on your own time. Serious long-term illness will be dealt with on an individual basis. In these circumstances, please contact your instructor to discuss appropriate arrangements.

Assignments:

Students are encouraged to discuss issues when working on assignments; however, you are expected to submit your own work for grading (assignments are individual work). There is a fine line between cooperating with your colleagues (discussing problems and ideas) and copying solutions (plagiarism). Not only plagiarism is 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 online on cuLearn. When submitting assignments, double check that your material has indeed been submitted.

Late assignments will be graded according to the following policy: a cumulative 10% penalty per day (i.e., 24 hours) with a maximum of two days.

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 see the Student Guide (https://carleton.ca/equity/?p=191).

**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 see the Student Guide.
**Academic Accommodations for Students with Disabilities:** The Paul Menton Centre for Students with Disabilities (PMC) provides services to students with Learning Disabilities (LD), psychiatric/mental health disabilities, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), chronic medical conditions, and impairments in mobility, hearing, and vision. If you have a disability requiring academic accommodations in this course, please contact PMC at 613-520-6608 or pmc@carleton.ca for a formal evaluation. If you are already registered with the PMC, contact your PMC coordinator to send me your Letter of Accommodation at the beginning of the term, and no later than two weeks before the first in-class scheduled test or exam requiring accommodation (if applicable). After requesting accommodation from PMC, meet with me to ensure accommodation arrangements are made. Please consult the PMC website for the deadline to request accommodations for the formally-scheduled exam (if applicable).

**Health and safety:**

**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.

**Tentative Week-by-Week Outline**
The following is a tentative outline of the course; it might change, based on time constraints:
Week 1: Introduction to Software Engineering. 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 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: Requirement Elicitation (continued).
Week 4: Object Oriented Analysis using UML. Producing an analysis model that the developers can unambiguously interpret. Formalizing the requirements (requirement elicitation) into specifications (Analysis).
Week 5: Object Oriented Analysis (continued).
Week 6: Object Oriented Analysis (continued).
Winter Break. Classes are suspended.
Week 7: Object Oriented Analysis (continued).
Week 8: System Design. 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 9: System Design (continued).
Week 10: System Design (continued).
Week 11: Design Patterns.
Week 12: Verification and Validation.