CARLETON UNIVERSITY

Department of Systems and Computer Engineering

SYSC 3020 **Introduction to Software Engineering** **Summer 2016**

Course Handout

Lecturer:

Dr. Greg Franks, Room 5204CB, Ph: 520-2600 ext 1749, greg@sce.carleton.ca

Course 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, software construction (i.e., 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 have not been 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:

1. 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;
2. To understand the main principles that underlie these phases;
3. To master model-based software development by using the UML;
4. To understand the challenges of software evolution;
5. To understand the challenges of software verification and validation.

And the three main objectives of this course are:

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

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.

Prerequisites:

SYSC 2004 and SYSC 2006. Precludes additional credit for SYSC 3120 and COMP 3004. Students must satisfy the prerequisites in order to remain registered in the course. Students who have not completed the prerequisites are required to either withdraw from the course or submit a prerequisite waiver online at `http://www.sce.carleton.ca/ughelp`. Students not meeting these conditions will be deregistered from the course after the last day for course registration.

Textbook:

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

References:

Referenced: G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, 2nd ed., Addison-Wesley, ISBN 0-321-26797-4, 2005.

Grading Scheme:

25% Assignments (3) and labs (6)
25% Mid-Term Exam (TBD, In Class)
50% Final Examination (Scheduled, 3 hours, no calculators, hand-written aid sheet).

Important Notes:

1. **This is a compressed course and finishes after six weeks! Deadlines will come up really quickly!**
2. Students **must pass the Final Examination Paper** (50% or higher) in order to pass the course. (i.e. Failing to pass the Final Examination results in an F grade for the course).
3. The final exam will be held during the formal examination period set out in the University Calendar and will be scheduled by Exam Services. I will not accomodate any special requests or alternate arrangements. For SYSC 3020, **only students who have attempted five out six labs** are eligible to write the final examination or, where circumstances warrant, apply to the Registrars Office for deferral of the final exam. Attempted means that the TA has seen (and in most cases graded) your work.
4. **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, the midterm weight will be added to the final examination weigh (i.e., the final exam becomes worth 75% of your final mark). (See the Undergraduate Calendar for regulations on deferred term work).
5. **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.
6. **Students who have previously taken SYSC 3020.** Please note that there are no lab exemptions from this course.

Academic Accomodation

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 week of class, or as soon as possible after the need for accommodation is known to exist. For more details see the Student Guide to Academic Accomodation.

Religious obligation: write to me with any requests for academic accommodation during the first week of class, or as soon as possible after the need for accommodation is known to exist. For more details see the Student Guide to Academic Accomodation.

Students with disabilities requiring academic accommodations in this course must register with the Paul Menton Centre for Students with Disabilities (PMC) for a formal evaluation of disability-related needs. Documented disabilities could include but are not limited to mobility/physical impairments, specific Learning Disabilities (LD), psychiatric/psychological disabilities, sensory disabilities, Attention Deficit Hyperactivity Disorder (ADHD), and chronic medical conditions. Registered PMC students are required to contact the PMC, 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 only require accommodations for your formally scheduled exam(s) in this course, please submit your request for accommodations to PMC by the deadlines published on the PMC website.

Course Outline:

- Week 1a: Introduction to Software Engineering. The nature of software, history and scope of software engineering, relationships with other fields, fundamental principles, software life cycle.
- Week 1b: 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 2a: Requirement Elicitation continued.
- Week 2b: Object Oriented Analysis using UML. Producing an analysis model that the developers can unambiguously interpret. Formalizing the requirements (requirement elicitation) into specifications (Analysis).
- Week 3a: Object Oriented Analysis (continued)
- Week 3b: Object Oriented Analysis (continued)
- Week 4a: Object Oriented Analysis (continued)
- Week 4b: 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. (**Midterm**).
- Week 5a: System Design
- Week 5b: System Design.
- Week 6a: Design Patterns.
- Week 6b: Verification and Validation.