Carleton University

Department of Systems and Computer Engineering

SYSC 3010	***Systems Development Laboratory***	***Fall 2013***

Course Outline

Lecturer

C. Schramm, Office: 6202CB, Email: schramm@sce.carleton.ca

Calendar Description:

Development of expertise in designing, implementing and testing industrial-quality, reusable software through team projects. Applying modern programming languages, design patterns, frameworks, UML, and current development processes (refactoring, iterative and incremental development, version control techniques) to medium-scale projects; for example, embedded or mobile applications.

Course Objectives

- Students will learn processes and tools for working effectively in teams.
- Students will be able to use key software patterns for both system architecture and software design.
- Students will describe concepts of iterative and incremental systems development processes
- Students will be able to create unit tests, at both the architectural level and the class level.
- Students will be able to construct moderately complex systems composed both of embedded computing and web-enabled applications.
- Students will be able to extend and integrate existing software utilities in well-known frameworks, such as (but not limited to) Java's Swing and xUnit.
- Students will be able to communicate their design effectively to their peers in both oral and written forms in a collaborative and comparative environment.
- Students will work cooperatively in groups, scheduling their own work within the time allocated and considering relevant design aspects as safety, performance, cost and product life cycle
- Reflect weekly on the work and challenges encountered

Carleton University

Department of Systems and Computer Engineering

SYSC 3010 ***Systems Development Laboratory*** ***Fall 2013***

Course Outline

Course Outcomes

This course is an introduction to the software development life cycle and to team project management. From a technical perspective the course is expected to draw from material covered in prior and concurrent courses, with lectures largely providing additional depth or considerations regarding scale and complexity. Complementary to the technical aspects and with equal importance are the challenges associated with working with a team to produce a project on schedule. Resulting from this course a student should see the technical importance of topics already covered in the program, have broader appreciation of the design process and aspects such as teamwork, discipline, scheduling and communication. Self reflection is a key aspect of professionalism and this is encouraged and examined within the course. The expected result will be a deeper appreciation of the importance of technical knowledge, the design cycle and professional skills which will be beneficial for, co-op placements, the fourth year project and final employment.

.

Prerequisites

SYSC 2100 and SYSC 2004, and third-year status in Computer Systems Engineering are the course prerequisites. Students who have not satisfied these prerequisites must either: a) withdraw from the course, b) submit a prerequisite waiver online at www.sce.carleton.ca/ughelp, or c) will be deregistered from the course after the last day to register for courses in the current term.

Instructional Resources.

Course material, project requirements and lab exercises will be posted on the course website through CULearn.carleton.ca.

There is no course textbook designated for this course. A bibliography of reference books that are relevant to this course will be posted on the course Web site.

Lecture and Lab Periods

There is one two-hour lecture per week.

Course Outline

Labs are held 3 hours a week. Labs are mandatory for your learning, yet will not contribute to your final grade. Lab hours will be used to introduce tools and practice material that was introduced during class. The lab exercises will need to be completed and submitted before the end of the lab. The labs will also be used for ½ hour team meetings, observed or attended by the instructor and/or TA(s). Minutes of these meetings must be taken and submitted as part of the term project.

Project

A major component of the course is a project that will lead you through the process of building a reasonably complex system. This will be a group project. Each team member must participate in all aspects of the project: design, coding, testing and debugging, etc. The project will be of the team's own idea, yet must meet the universal technical requirements posted for the course. With help, the teams will plan the project as a series of milestones that progress incrementally and iteratively through the engineering lifecycle. The group's work will be evaluated at two points using the same rubric used for Fourth Year Projects, and feedback will be given. The final grade for the project will incorporate marks for both the "process" (how the group functioned over the term) and the "product" (the quality of the final system delivered). Individual marks will be given to each student reflecting their contribution as evaluated both through self and peer assessment. Groups will present their work and defend their work, in mid-term and at the end of term. Each student will also be required to maintain a reflection journal throughout the term, documenting the major design or implementation lessons experienced.

Evaluation and Grading Scheme

The final grade will be calculated as follows:

Trial Project	10%
Project Proposal	10%
Project Progress Report & Presentation	25%
Project Final Report & Presentation	25%
Technical Assessment of Project	20%
Reflection Journal	10%

Failure to produce a Final Project or a Reflection Journal can result in a failing grade.
Most of the work will be done in groups, so that there will be only one group report and a joint group presentation that will be graded.
Individuals will be awarded separate marks depending on their contribution.

Special Arrangements

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 your instructor 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://www2.carleton.ca/equity/

Religious obligation: write to your instructor 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://www2.carleton.ca/equity/

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 your instructor to ensure accommodation arrangements are made. Please consult the PMC website for the deadline to request accommodations for the formally-scheduled exam (*if applicable*) at http://www2.carleton.ca/pmc/new-and-current-students/dates-and-deadlines/

You can visit the Equity Services website to view the policies and to obtain more detailed information on academic accommodation at http://www2.carleton.ca/equity/”

Course Outline

Health and Safety:

Every student should have a copy of our Health and Safety Manual. An electronic version of the manual can be found at http://www.sce.carleton.ca/courses/health-and-safety.pdf

Lecture Schedule (Tentative)

1. Course Introduction, Trial Project Introduction, Embedded System Introduction
2. A Tour of the Embedded System, including network and communication
3. A Working Introduction to UML. TeamWork: Roles, Meetings, Minutes, Processes, Communication, Peer Assessment, Self-Assessment, Tools (SVN, CVS, google, GIT), Documents (Design, Test) Design Process
4. Design Process, Trial Project Symposium , Process: Iteration and Incremental, Testing, Architectural Design, Project Roundtable: Collaborative Development of (Incremental)Milestones
5. Design Principles: Architectural or System Design, Communication Protocol Design. Case Study: Manufacturing Work Cell (FSM).
6. Design Patterns: Event Model (Callbacks, Observer).
7. (Unit) Testing: xUnit Testing Frameworks: xUnit, JUnit, STAF. Roundtable: Collaborative Development of Project Tests.
8. Design Pattern: MVC. Principles of GUI Design. Strategies GUI Testing.
9. Framework: Persistence. Using XML (Files)
10. Automation of Testing through Scripts.
11. Refactoring Strategies
12. Integration or Acceptance Testing. Handling Last Minute Changes.
13. Review