Course Objectives

- Students will be able to use industry-standard 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 communicate their design effectively to their peers in both oral and written forms in a collaborative and comparative environment.
- Students will be able to 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.
- Students will be able to reflect on the work and challenges encountered.

Learning 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 their degree 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.

Course Web Site

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

Textbook and References

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

Evaluation and Marking Scheme

The final grade will be calculated as follows:

**Individual Assessments** - 40%

- Technical Memo: 15%
- Lab Work: 10%
- Class Peer Reviews (3): 9%
- Code Review: 6%

**Team Assessments** - 60%
Failure to work in a group 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 joint group presentation that will be graded. When needed, individuals will be awarded separate marks commensurate with their contribution, including zero.

**Labs**

There is one two-hour lecture per week. Lectures are mandatory, some of the lectures involving class-wide exercises that will be graded.

There is one three-hour lab period per week. Labs are mandatory for your learning, and full attendance for the entire 3-hour period is demanded.

- During the first section of term, there will be a lab exercise associated with the current lecture. The lab exercise is to be completed individually and/or in pairs, and must be submitted before the end of the lab period for grading.
- During the remainder of the term, the labs are either:
  - Open work time for your group’s term project – while there is no explicit deadline, attendance is mandatory. It is a time when all members of the group must be available, and where the TAs and instructor can check in with interim progress.
  - Milestone demonstrations by the group to the instructor and TAs – if a team member is missing from the demo, that person receives zero for the demonstration.

**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 guidance, 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 several points, 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.

Exams

The final examination is for evaluation purposes only and will not be returned to students. You will be able to make arrangements with the instructor or with the department office to see your marked final examination after the final grades have been made available.

General Regulations

- **Copyright on Course Materials**: The materials created for this course (including course outline, slides, posted notes, labs, project, assignments, quizzes, exams and solutions) are intended for personal use and may not be reproduced or redistributed or posted on any web site without prior written permission from the author(s).
- **Attendance**: Students are expected to attend all lectures and lab periods. The University requires students to have a conflict-free timetable. For more information, see the current Undergraduate Calendar, Academic Regulations of the University, Section 1.2, Course Selection and Registration and Section 1.5, Deregistration. Requests to accommodate a missed midterm exam, lab periods, etc., because of conflicts with jobs or vacation plans will not be considered.
- **Health and Safety**: Every student should have a copy of our Health and Safety Manual. A PDF copy of this manual is available online: http://sce.carleton.ca/courses/health-and-safety.pdf.
- **Deferred Term Work**: Students who claim illness, injury or other extraordinary circumstances beyond their control as a reason for missed term work are held responsible for immediately informing the instructor concerned and for making alternate arrangements with the instructor and in all cases this must occur no later than three (3.0) working days after the term work was due. The alternate arrangement must be made before the last day of classes in the term as published in the academic schedule. For more information, see the Academic Regulations of the University, Section 2.6, Deferred Term Work.
- **Appeal of Grades**: The processes for dealing with questions or concerns regarding grades assigned during the term and final grades is described in the Academic Regulations of the University, Section 2.7, Informal Appeal of Grade and Section 2.8, Formal Appeal of Grade.
- **Academic Integrity**: Students should be aware of their obligations with regards to academic integrity. Please review the information about academic integrity at: https://carleton.ca/registrar/academic-integrity/ This site also contains a link to the complete Academic Integrity Policy that was approved by the University’s Senate.
- **Academic Accommodations**: Requests for 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
    Please contact 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: carleton.ca/equity/wp-content/uploads/Student-Guide-to-Academic-Accommodation.pdf
  - Religious obligation
    Please contact 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: carleton.ca/equity/wp-
Academic Accommodations for Students with Disabilities
If you have a documented disability requiring academic accommodations in this course, please contact the Paul Menton Centre for Students with Disabilities (PMC) at 613-520-6608 or pmc@carleton.ca for a formal evaluation or contact your PMC coordinator to send your instructor your Letter of Accommodation at the beginning of the term. You must also contact the PMC 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 as soon as possible to ensure accommodation arrangements are made. carleton.ca PMC

Survivors of Sexual Violence
As a community, Carleton University is committed to maintaining a positive learning, working and living environment where sexual violence will not be tolerated, and is survivors are supported through academic accommodations as per Carleton's Sexual Violence Policy. For more information about the services available at the university and to obtain information about sexual violence and/or support, visit: carleton.ca/sexual-violence-support

Accommodation for Student Activities
Carleton University recognizes the substantial benefits, both to the individual student and for the university, that result from a student participating in activities beyond the classroom experience. Reasonable accommodation must be provided to students who compete or perform at the national or international level. Please contact 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. https://carleton.ca/senate/wp-content/uploads/Accommodation-for-Student-Activities-1.pdf

Tentative Week-By-Week Schedule

1. Course Introduction, Team Building & Trial Project Introduction, Embedded Hardware Introduction
2. Distributed Systems - Networking and JSON
3. I/O Systems - Sensors and Actuators
4. Term Project Proposals
6. Project Tools - Version Control Systems. Introduction to Design Patterns: Event Model (Callbacks, Observer, Model-View-Controller)
7. Term Project Design Reviews
8. Introduction to Android through Design Patterns
9. Persistence (Properties, Relational, SQL)
10. Unit and Integration Testing: xUnit Testing Frameworks: xUnit, JUnit, STAF. Collaborative Development of Project Tests.
11. Embedded Design – Solving Problems through FSMs
13. Refactoring Strategies using core Design Principles (Coupling, Cohesion, Demeter)
14. Term Project Code Review
15. Term Project Poster Session