



Carleton
UNIVERSITY

Department of
**Systems and
Computer Engineering**

SYSC 3110

Software Development Project

Calendar description

Development of expertise in designing, implementing and testing maintainable, reusable software through team projects. Applying modern programming languages, design patterns, frameworks, UML and modern development processes (detection of olfactible source code defects, refactoring, iterative and incremental development, version control techniques) to medium-scale projects.

Includes: Experiential Learning Activity.

Lectures two hours a week, laboratory three hours a week.

<http://calendar.carleton.ca/undergrad/courses/SYSC/>

Prerequisites

SYSC 2004 and SYSC 2100, and enrolment in Software Engineering.

Precludes additional credit for SYSC 2101, SYSC 3010 and COMP 2404.

Prior knowledge

The course requires basic knowledge in object-oriented programming (SYSC 2004) and a solid understanding of data structures and algorithms (SYSC 2100) to get into the more advanced topic of designing maintainable software.

Course objectives

Development of expertise in designing, implementing, and testing industrial-quality, reusable code through individual and team projects. Applying and extending previously acquired knowledge of patterns, frameworks, UML, iterative and incremental development, to medium and large-scale systems.

List of topics

- User Interface Design as an Exercise in Using Design Patterns.
- Part II Design in Practice: Refactoring and Test-driven development.
- Part III Framework Studies: JUnit, Persistence and possibly others.
- Lab hours will be used to introduce Eclipse as an Integrated Development Environment and to practice material introduced in class.

Learning outcomes

By the end of this course, students should be able to:

- Design software.
- Write code that applies a given design.
- Pick the appropriate data structure and algorithm and implement them.
- Draw simple UML class and interaction diagrams to capture and document design decisions.
- Test software.
- Detect code “smells”.
- Have some early experience of working in teams.

Graduate Attributes (GAs)

The Canadian Engineering Accreditation Board requires graduates of engineering programs to possess 12 attributes at the time of graduation. Activities related to the learning outcomes listed above are measured throughout the course and are part of the department’s continual improvement process. Graduate attribute measurements will not be taken into consideration in determining a student’s grade in the course. For more information, please visit: <https://engineerscanada.ca/>.

Graduate Attribute	Learning outcome(s)
1.4.S: Knowledge Base: Applied: Programming and algorithms	3
1.8.S: Knowledge Base: Developed: Software engineering	5
4.4: Design: Developed: Design solution(s)	1
4.5: Design: Developed: Design implementation / task(s) definition	2
5.1: Use of Engineering Tools: Developed: Diagrams and engineering sketches	4
6.1: Individual and Team Work: Developed: Personal and group time management	7
6.2: Individual and Team Work: Developed: Group culture, group dynamics	7
6.3: Individual and Team Work: Developed: Leadership: initiative and mentoring, areas of expertise, and interdisciplinary teams	7

Accreditation Units (AUs)

For more information about Accreditation Units, please visit: <https://engineerscanada.ca/>.

The course has a total of 44 AUs, divided into:

- Engineering Science: 50%
- Engineering Design: 50%

Instructor and TA contact

Specific to course offering (tbd)

Textbook (or other resources)

Specific to course offering (tbd)

Evaluation and grading scheme

Specific to course offering (tbd)

Breakdown of course requirements

Specific to course offering (tbd)

Tentative week-by-week breakdown

Specific to course offering (tbd)

General regulations

Specific to course offering (tbd)