Carleton UNIVERSITY | Department of Systems and Computer Engineering

SYSC 4120
Software Architecture and Design

Calendar description

Introduction and importance of software architectures and software system design in software engineering. Current techniques, modeling notations, methods, processes and tools used in software architecture and system design. Software architectures, architectural patterns, design patterns, software qualities, software reuse.

Includes: Experiential Learning Activity.

Lectures three hours a week, laboratory three hours alternate weeks.

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

Prerequisites

SYSC 3120.

Precludes additional credit for SYSC 3020, SYSC 4800 and COMP 3004.

Prior knowledge

Students should:

- Understand the basic principles of requirement engineering – requirement elicitation, analysis, and simple design.
- Use effectively Unified Modeling Language (UML) – use case modeling, object interactions, class diagrams, and state machine.
- Understand of static modeling vs dynamic modeling.
- Have an elementary understanding of the application of formal methods to systems design.
- Program in Java and/or C++ or any other high level OO program language.

Course objectives

This course focuses on software systems design, software architecture, and object design. The objectives are:

- Study software system design activities using architectural and design patterns, and the role of object design in software development.
- Master model-based software system design by using the UML and formal methods (e.g. OCL, LTL, 1st Order Logic).

List of topics

- Introduction to Software System Design
- System Design using UML and System Architecture
- Definition and objectives, object-oriented design with UML, architectural design, detailed design, concurrent software, safety analysis and fault tolerance
- Object Design
- Optimizing software architecture, optimizing class diagram
- Software Design Patterns
- Revisiting Design Patterns and Applications
- Other Software Engineering Issues
- Fault tolerance, real-time systems, verification and validation, re-engineering

Learning outcomes

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

- Use domain knowledge to identify, formulate, analyze complex software systems requirement.
- Clarify systems requirement definition and to deal with missing and/or badly-defined requirements.
- Make reasonable assumption about systems requirements and design a solution that best match the requirements.
- Conduct complex systems design using appropriate engineering techniques and tools (e.g. UML, LTL, OCL, 1st Order Logic, etc.). Specifically, UML and OCL will be used in this course because LTL and 1st Order Logic were introduced in SYSC 3120.
- Measure, verify, and validate the end product (i.e. the software system).

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)
1.8.S: Knowledge Base: Applied: Software Engineering	1
4.1: Design: Applied: Clear design goals	3
4.2: Design: Applied: Detailed design specifications and requirements	2
4.4: Design: Applied: Design solution(s)	4
4.5: Design: Applied: Design implementation / Task(s) definition	3, 4
4.6: Design: Applied: Alternate solution(s) definition	5
4.7: Design: Applied: Evaluation based on engineering principles	1

Accreditation Units (AUs)

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

The course has a total of 46 AUs, divided into:

- Engineering Science: 30%
- Engineering Design: 70%

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)