Carleton UNIVERSITY | Department of Systems and Computer Engineering

SYSC 2010
Programming Project

Calendar description

Programming, testing, and debugging of small team-based software projects that use data from sensors to display results graphically. Modern programming tools: frameworks, libraries, version control, package management, tool chains. Sensors, signal acquisition, display, and basic filtering. Introductory network programming.
Includes: Experiential Learning Activity
Precludes additional credit for SYSC 3010, SYSC 3110.

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

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

Prerequisites

Prerequisite(s): 2nd year status in Biomedical and Electrical Engineering or Communications Engineering.

Students who have not satisfied the prerequisites for this course must either withdraw from the course or obtain a prerequisite waiver by visiting the Engineering Undergraduate Academic Support Office.

Other Requirements/Program Information
For Biomedical and Electrical Engineering Students: SYSC 2010 was recently added to the Undergraduate Calendar. The first offering (trial offering) of the course will be Winter 2021 and will be an elective option for Biomedical and Electrical Engineering students. Students wishing to take the course will be permitted to count it towards ELEC 3908 or SYSC 2004 elective option in their program.

For Communications Engineering Students: SYSC 2010 was recently added to the Undergraduate Calendar. The first offering (trial offering) of the course will be Winter 2021 and will be an elective option for Communications Engineering students. Students wishing to take the course will be permitted to count it towards their Engineering Elective in their program.

Prior knowledge

Students should:
• Understand the concepts of software development as an engineering discipline.

• Understand the concepts of engineering problem solving, defining problems, designing solutions, using procedural programming.
• Have an introductory knowledge of an imperative programming language (Python or C)

Course objectives

By the end of this course students should:

- Software development skills
    - Have gained extensive experience designing and implementing programs using the Python language
    - Software development practices and tools including GIT, SDE, libraries and Agile.
    - Knowledge and ability to work with varied data types and structures
    - Programming concepts including an introduction to aspects of real-time programming
- Sensors and Signals skills
    - Understanding and experience with sensor devices and the signals they generate
    - Experience working with signals such as visualization, impairments, introduction to sampled signals and data flows from sensors.
    - Introduced to signal manipulation techniques including basic filtering and analysis.
- Team work and collaboration skills
    - Understand and experience collaborative software development through work within a team including project/task management
- Engineering applications
    - Applications of sensors within a Biomedical assessment use case

List of topics

- Tools and Methods
    - Software
    - Development (tools)
    - Intro to Agile methods
    - Team creation and working models
- Signals and Sensors
    - Signals introduction – what is a signal? Visualizing in real-time
    - Signals – impairments
    - Filtering (why, design method)
    - Sensors: What is a gyro and accelerometer as a sensor.
    - Analysis introduction – signal identification (why, applications, techniques)
    - Biomedical Applications
    - Some biomedical signals and example applications
    - The n-back test

- Software Design
    - Libraries for Python Licensing implications
    - Data from sensors (JSON v CSV v other)
    - Intro to Real-Time concepts (dealing with a signal as it arrives)
    - Reporting and program outputs (console vs file vs DB)
    - Programming models – Embedded vs PC based vs cloud

Learning outcomes

1. Apply software design tools and techniques to solve a specific engineering application
2. Design and Create small scale software applications within a team
3. Apply basic signal processing and analysis techniques to a sensor signal
4. Apply imperative programming and problem solving to engineering applications.
5. Understand and gain insight into the nature software development in industry.

Graduate Attributes (GAs)

The Canadian Engineering Accreditation Board requires graduates of engineering programs to possess 12 attributes at the time of graduation. There are no GAs assigned to this course. For more information, please visit: https://engineerscanada.ca/.

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)

Important Information

Specific to course offering (tbd)