Carleton UNIVERSITY | Department of Systems and Computer Engineering

ECOR 1042
Data Management

Calendar Description

Software development using container data types (sequences, sets, maps) for data management. Modules. Data files. Incremental, iterative development of programs. Introduction to designing and implementing numerical algorithms.

Lectures three hours per week, laboratories three hours per week.

Prerequisites

ECOR 1041 with a minimum grade of C-; MATH 1004 (may be taken concurrently).

This course may not be taken concurrently with ESLA 1300 or ESLA 1500.

Precludes additional credit for COMP 1005, COMP 1405, ECOR 1051, ECOR 1606, SYSC 1005.

Prior Knowledge

Basic experience in computer programming with Python is required. The background we assume is:

- *language*: reasonable proficiency in reading and writing English;
- *math*: understanding of integers and operations on integers; understanding of functions as mappings from one set (the domain) to another set (the codomain);
- *math (2)*: basic understanding of limits, differentiation of the elementary functions, rules of differentiation, linear algebra, curve sketching, and maximization/minimization.
- *logic*: familiarity with logical *and*, *or* and *not*;
- *computer literacy*: ability to use email, browse the World Wide Web, and edit text files.
- *Python*: know how basic Python functions, statements, and operators work, including *min, max, int, float, str, bool, abs, ceil, floor, print, get, append, update, range, pow, in, for, while, if, elif, else, def, return, , +, -, *, **, /, //, %, [],list, , , etc*.

Course Objectives

By the end of this course, students should:

- know the fundamentals concepts of procedural programming, using Python as the programming language.
- have gained practical experience using lightweight, modern software engineering practices in a team environment to design and implement small-scale programs.

- have developed a "mental model" of computation; in other words, learned how to reason about and visualize the execution of program code.
- understand the use of software experiments as an aid to learning.
- know how to implement numerical algorithms as functions in Python.

List of Topics

1. Container data types in Python: sets, tuples, dictionaries (maps). Creating objects, operators, built-in functions, and methods.
2. Defining and importing modules.
3. Incremental software development.
4. Reading and writing text files.
5. Testing, in the context of a project.
6. Interactive and batch user interfaces.
7. Introduction to numerical algorithms: algorithms for root finding, sorting elements, one-dimensional optimization and curve fitting.

Learning Outcomes

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

1. apply all Python concepts learned in ECOR1041, sets, tuples, dictionaries and modules to develop small scale programs including: *min, max, int, float, str, bool, abs, ceil, floor, print, get, append, update, range, pow, in, for, while, if, elif, else, def, return, import, +, -, *, **, /, //, %, [], {}, list, set, tuple, and dictionary.*
2. know the "client-side" view of four Python containers for organizing data, namely: lists, tuples, sets and dictionaries.
3. select and use appropriate container(s) in their programs.
4. test functions using a simple unit-testing framework.
5. apply simple debugging techniques (e.g., inserting print statements to instrument code or by tracing the code using a program visualization tool) to locate faults in their code.
6. work in a small team to iteratively and incrementally design, implement and test a small-scale, interactive program that is partitioned into multiple modules, given a detailed specification of the functional requirements.
7. apply Python concepts to implement numerical algorithms as functions to solve mathematical problems.

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)
1.3 Knowledge base for engineering: Fundamental engineering concepts	all
6 Individual and team work: 6.1 Personal and group time management 6.2 Group culture, group dynamics 6.3 Leadership: initiative and mentoring, areas of expertise, and interdisciplinary teams	6
7.1 Communications skills: Giving and following instructions	6&7 (following), 6 (giving)

Accreditation Units (AUs)

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

The course has 27 AUs divided into:

- Engineering Science: 100%
- Engineering Design: 0%

Instructor and TA contact

Specific to course offering (tbd)

Textbook and Software

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)