Carleton UNIVERSITY | Department of Systems and Computer Engineering

ECOR 1041
Computation and Programming

Calendar Description

Software development as an engineering discipline, using a modern programming language. Language syntax and semantics. Tracing and visualizing program execution. Program style and documentation. Testing and debugging tools and techniques.

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

Prerequisites

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

No prior experience in computer programming is required. The only 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).
- *logic*: familiarity with logical *and*, *or* and *not.*
- *computer literacy*: ability to use email, browse the World Wide Web, and edit text files.

Course Objectives

The course introduces students to concepts that are central to understanding computation and teaches students how to design, code, test, and debug small-scale programs written in the Python programming language. By the end of this course, students should:

- know how integers and floating-point numbers are represented on a computer.
- know the fundamental concepts of procedural programming, using Python as the programming language.
- 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.

List of Topics

1. Introduction to Python: using the Python shell as a calculator. Numeric types (`int` and `float`); operations supported by these types. Construction and evaluation of expressions.
2. The binary number system. Representation of integers and floating-point numbers in binary. Limitations of real-number representation in binary.
3. The assignment operator: binding values to variables. Using variables in expressions. Using Python Tutor to visualize assignment operations.
4. Calling built-in functions from the shell. Importing functions from modules. Evaluating call expressions.
5. Defining functions. Passing arguments to and returning values from functions. Using the shell to test functions interactively. Using Python Tutor to trace and visualize the execution of function definitions (creation of function objects), function calls, and function execution. Using local variables in function definitions.
6. Simple programs (scripts). The `print` function. Functions that do not have return statements.
7. A step-by-step recipe for designing functions.
8. Using a simple unit-testing framework to automate testing.
9. Character strings (type `str`). Interactive programs (scripts): the input function.
10. Boolean values (type `bool`). Boolean operators. Relational operators. Evaluating expressions that have numbers, boolean and relational operators. Comparing strings. Making decisions: (`if, if-else, if-elif-else`, and `if-elif` statements).
11. Lists: Python's `list` type; creating list objects; list operators, built-in functions, and methods. Using a `for` loop to iterate over a list. Using the `range` function to generate a sequence of list indices.
12. Repeating actions until a condition is satisfied (`while` statements).

Learning Outcomes

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

1. Represent signed and unsigned integers and decimal numbers using binary representation; explain the limitations of real-number computation using floating-point arithmetic.
2. Evaluate expressions consisting of literal values, variables, and operators, using the same evaluation rules as Python. In other words, predict the values that Python calculates when it evaluates expressions.
3. Design and implement Python programs that include Python's lists and fundamental constructs for controlling program execution to solve a given problem. The constructs include: (1) sequential execution as determined by the ordering of executable statements; (2) selection (`if, if-else`, and `if-elif-else` statements); (3) repetition (`for` and `while` statements); (4) function calls.

4. Trace short Python programs. In other words, explain what happens, step-by-step, as the computer executes each statement

5. Visualize Phyton code execution; in other words, draw diagrams that depict the variables in the program's global frame and function activation frames (function arguments and local variables), and the objects that are bound to the variables.

6. Design, implement, execute, trace and test simple functions using Python. Testing should be done using the Python shell and simple unit-testing framework.

7. Debug a program using simple debugging techniques (e.g., inserting print statements to instrument code or by tracing the code using a program visualization tool)

8. Explain the "client-side" view of one Python container for organizing data, namely: lists

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

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%

Special Information for Pandemic Measures

- All members of the Carleton community are required to follow COVID-19 prevention measures and all mandatory public health requirements (e.g. wearing a mask, physical distancing, hand hygiene, respiratory and cough etiquette) and mandatory self-screening prior to coming to campus daily.

- If you feel ill or exhibit COVID-19 symptoms while on campus or in class, please leave campus immediately, self-isolate, and complete the mandatory symptom reporting tool. For purposes of contact tracing, attendance will be taken in all classes and labs. Participants can check-in using posted QR codes through the cuScreen platform where provided. Students who do not have a smartphone will be required to complete a paper process as indicated on the COVID-19 website.

- All members of the Carleton community are required to follow guidelines regarding safe movement and seating on campus (e.g. directional arrows, designated entrances and exits, designated seats that maintain physical distancing). In order to avoid congestion, allow all previous occupants to fully vacate a classroom before entering. No food or drinks are permitted in any classrooms or labs.
- For the most recent information about Carleton's COVID-19 response and required measures, please see the University's COVID-19 webpage and review the Frequently Asked Questions (FAQs). Should you have additional questions after reviewing, please contact covidinfo@carleton.ca
- Please note that failure to comply with University policies and mandatory public health requirements, and endangering the safety of others are considered misconduct under the Student Rights and Responsibilities Policy. Failure to comply with Carleton's COVID-19 procedures may lead to supplementary action involving Campus Safety and/or Student Affairs.

Instructor and TA contact

Section S: Dr. Wafa Hasanain; Email: wafahasanain@cmail.carleton.ca

Office hours and TA information are posted on Brightspace.

Textbook and Software

Textbook

Practical Programming (Third Edition): An Introduction to Computer Science Using Python 3.6, Paul Gries, Jennifer Campbell, Jason Montojo, Pragmatic Bookshelf, 2017, ISBN-13: 978-1-68050-268-8.

Despite the phrase "Python 3.6" in the title, the book is equally applicable to the Python release we will be using.

Multiple eBook formats and/or a paperback copy of this book can be purchased directly from the publisher's website: pragprog.com. *Make sure you buy the third edition, not the second edition.* eBooks in multiple formats, as well as print copies, are available.

A **free** digital copy of the course textbook is available through O'Reilly (a digital library) and can be accessed using the following steps:

1. Go to https://library.carleton.ca/find/databases/oreilly
2. Click on the "Connect: O'Reilly" hyperlink
3. You will be auto-magically signed into the database
4. At the very top of the page (to the right of the O'Reilly logo in red) is a search box
5. Type in "Practical Programming Jennifer Campbell" and hit ENTER
6. Select the first option: "Practical Programming, 3rd Edition."
7. You can then navigate the Table of Contents to access any textbook section!

Software

We will be using Python in this course, specifically the Wing 101 Python IDE. The software is free and available for the Windows, macOS, and Linux operating systems. Please refer to Brightspace to install the various software packages required for the labs and tests.

You are welcome to use other Python IDEs (e.g., PyCharm), but it is your responsibility to ensure that the code you submit will run in Wing 101 without errors. If the TA experiences any trouble during marking, you will receive zero for your submission.

Learning how to trace and explain the execution of short programs is an important learning outcome in this course. We will use Python Tutor (pythontutor.com) during course lectures, a free Web-based tool that helps us visualize what happens as the computer executes each line of a program's source code, step-by-step.

Evaluation and Grading Scheme

Students will be evaluated using reading quizzes, laboratory work, a midterm exam, and a final exam.

To pass the course (i.e., achieve at least C-), students must pass the final examination (i.e., obtain at least 25% out of 50%) **or** (2) obtain a minimum of 35% out of 70% when adding the midterm and final exam.

For students who satisfy the above criteria, numeric marks out of 100 will be calculated by weighting the course components as shown in this table:

Component	**Weight**
Reading quizzes	6%
Labs	24%
Midterm (Thursday, June 2^{ed} 8:35 – 9:55 am)	20%
Final exam (TBD)	50%

This mark will be converted to a letter grade, using the table of percentage equivalents shown in the *Undergraduate Calendar, Academic Regulations of the University, Section 5.4, Grading System.*

Note: Second-year status is a prerequisite for the second-year engineering, science, and mathematics courses that are part of your program. One of the requirements for achieving second-year status is a minimum grade of C- in each of ECOR 104x courses, including ECOR1041.

Breakdown of Course Requirements

Reading Quizzes

Reading assignments from the textbook will be posted to introduce the terminology and concepts applied in upcoming lectures and labs. Online short multiple-choice quizzes related to the reading assignments will be posted to test your reading comprehension and preparedness for the lecture. You will need to complete those quizzes by the date posted on Brightspace. **There will not be deferred reading quizzes or extensions.**

Labs

Labs provide short programming exercises that are intended to help you understand concepts that have been introduced in the lectures.

All labs should be completed during your lab section and submitted at the end of the lab. If you miss the submission deadline, your mark for the missed lab will be zero. The best 8 labs will count towards your grade. All labs are equally weighted.

There will not be deferred labs or extensions. For long-term illness, contact your instructor. Requests for extra time to work on a lab because of one-day illnesses or because you do not have the required files from a previous lab will not be approved.

Students are responsible for backing up their lab work; for example, we recommend copying your files to a USB flash drive or a cloud-based file hosting service, e.g., Google Drive, Dropbox, OneDrive, etc.

Important Considerations when Submitting the Labs

- It is the students' responsibility to ensure that all submissions are completed on time.
- Technical problems do not exempt you from the requirement to meet the submission deadlines. To avoid last-minute problems, you are advised to:
    - Periodically submit your "work in progress" as a draft; for example, submit the file(s) containing the work you have completed at least once an hour. Each submission before the deadline overwrites the previous version.
    - Submit your final submission for grading
    - Download the submitted file to ensure that you submitted the correct file.
- Make sure that each file you submit has (1) your name and student number, (2) the specified filename and format, and (3) runs without throwing errors. If these instructions are not followed, the following penalties will be applied:
    - Missing the name and student number:
        - Week 0: 10% penalty in the submission
        - Week 1 onwards: 25% penalty in the submission
    - File and extension name:
        - If the wrong file extension prevent your code from running properly: zero for the submission.
        - If the code runs properly:
            - Week 0: 25% penalty in the submission
            - Week 1 onwards: 50% penalty in the submission
    - The file runs with errors: zero for the submission

Exams (Midterm and Final)

The Midterm and Final exams will be held to assess your *individual* learning. Each test will cover **all material** covered up to that point in the course. Both the midterm and final exams will be **closed-book**. Detailed instructions about the midterm and final exams will be posted on Brightspace.

Failure to write one or more exams is cause for failure of the course.

Students who are unable to write the midterm exam because of illness or other circumstances beyond their control will write the original midterm for feedback, and the weight of the midterm will be moved to the final. Such students must contact their instructor to request this accommodation no later than three working days after the exam date, and the request must be fully supported by appropriate documentation (in cases of illness, a medical certificate is required). The University recognizes that medical certificates may be difficult to obtain during this time. For the Winter 2021 term, if you require accommodations because of illness, you can submit a self-declaration form in lieu of a medical certificate. The form can be obtained from the Registrar's Office website: https://carleton.ca/registrar/wp-content/uploads/self-declaration-1.pdf. For more information, see the Academic Regulations of the University, Section 4.4, Deferred Term Work.

The weight of the midterm will not be moved to the final because of poor performance.

Students who are unable to write the final exam because of illness or other circumstances beyond their control must pursue a formal application for a deferred through the Registrar’s Office.

For more information about deferred and missed exams, please see the *Academic Regulations of the University, Section 4.4, Deferred Term Work*.

Requests for accommodation because of poor performance on one or several exams/tests/labs will not be considered.

Tentative week-by-week breakdown

Lecture	**Topic**
1 – Week 0	Introduction, Data Types & Operators
2 – Week 1	Binary representation of numbers: integers and floats
3 – Week 1	Variables
4 – Week 2	Functions (Call)
5 – Week 2	Functions (Definition)
6 – Week 3	Function Design Recipe (FDR)
7 – Week 3	Strings
8 – Week 4	**Midterm** (Thursday, June 2ed 8:35 am to 9.55 am)
9 – Week 4	Conditionals
10 – Week 5	Booleans
11 – Week 5	Lists and for loops
12 – Week 6	While loops

Early Feedback

See Section 5.3 of the *Academic Regulations of the University*.

Your solutions to the lab exercises and midterm will normally be graded within one week after the submission deadline.

Course Web Site

This course uses Brightspace, Carleton's learning management system. To access your courses on Brightspace, go to brightspace.carleton.ca and log in with your MyCarleton1 username and password.

For help and support, go to https://carleton.ca/brightspace/students/

General regulations

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 2.1.3, Course Selection and Registration, and Section 2.1.7, Deregistration.*

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 current *Undergraduate Calendar, Academic Regulations of the University, Section 4.4, 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 *Undergraduate Calendar, Academic Regulations of the University, Section 3.3.4, Informal Appeal of Grade, and Section 3.3.5 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.

Plagiarism: Plagiarism (copying and handing in for credit someone else's work) is a serious instructional offense that will not be tolerated.

E-Proctoring: Please note that tests and examinations in this course will use a remote proctoring service provided by Scheduling and Examination Services. You can find more information at https://carleton.ca/ses/e-proctoring/. The minimum computing requirements for this service are as follows:
Hardware: Desktop or Laptop
OS: Windows 10, Mac OS 10.14, Linux Ubuntu 18.04
Internet Browser: Google Chrome, Mozilla Firefox, Apple Safari, or Microsoft Edge

Internet Connection (High-Speed Internet Connection Recommended)
Webcam (HD resolution recommended)
Note: Tablets, Chromebooks, and Smartphones are not supported at this time. Windows-based tablets are not supported at this time.

Academic Accommodation: You may need special arrangements to meet your academic obligations during the term. You can visit the Equity Services website to view the policies and to obtain more detailed information on academic accommodation at http://www.carleton.ca/equity/. For an accommodation request, the processes are as follows:

- **Pregnancy or 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, see https://carleton.ca/equity/wp-content/uploads/Student-Guide-to-Academic-Accommodation.pdf
- **Academic Accommodations for Students with Disabilities**: The Paul Menton Centre for Students with Disabilities (PMC) provides services to students with Learning Disabilities (LD), psychiatric/mental health disabilities, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), chronic medical conditions, and impairments in mobility, hearing, and vision. If you have a disability requiring academic accommodations in this course, please contact PMC at 613-520-6608 or pmc@carleton.ca for a formal evaluation. If you are already registered with the PMC, contact your PMC coordinator to send me your ***Letter of Accommodation*** at the beginning of the term and no later than two weeks before the first in-class scheduled test or exam requiring accommodation (*if applicable*). **Requests made within two weeks will be reviewed on a case-by-case basis.** After requesting accommodation from PMC, meet with me to ensure accommodation arrangements are made. Please consult the PMC website (www.carleton.ca/pmc) for the deadline to request accommodations for the formally-scheduled exam (*if applicable*).
- **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 where 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: https://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. For more details, see https://carleton.ca/senate/wp-content/uploads/Accommodation-for-Student-Activities-1.pdf

Copyright on Course Materials: The materials created for this course (including the course outline and any 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).