

Course Description:

A first course in programming emphasizing problem solving and computational thinking. Topics include algorithm, data types, conditionals, iteration, data structures, functions, objects, testing, sorting, searching, and run-time analysis.

Instructor: Farah Chanchary (she/her)

Email: farahchanchary@cunet.carleton.ca

Lectures: A - Tuesdays and Thursdays: 8:35 – 9:55 am.
B - Tuesdays and Thursdays: 11:35 – 12:55 pm.

Tutorials: A1 - Tuesday: 18:05 - 19:25 pm.
A2 - Monday: 18:05 - 19:25 pm.
B1 - Monday: 19:35 - 20:55 pm.
B2 - Tuesday: 18:05 - 19:25 pm.

Course Website: all course materials and resources will be available on this [course website](#).

Office Hours: Wednesday 11-1 pm (or by appointment) on Discord. TA's office hours and contact information are available on the course website.

Online platform for Q/As: All questions pertaining to lectures, tutorials, exams, and course material will be answered on Discord. Sign up information can be found on the website. Students are encouraged to post all course related questions to the appropriate Discord channel.

Learning Modality:

This is a blended (online) course. Lectures will be a combination of live classes (held and recorded during the scheduled class times) and pre-recorded lectures. Recorded live classes will be posted to the course webpage within 24 hours. Pre-recorded lectures will be posted 24 hours before the scheduled class time, and class time will be spent working through examples, live coding, class discussions and Q&A. We will use Zoom for all live classes (link will be available in Brightspace). Tutorials and Office hours will also be conducted online on our official Discord server (more information on course website).

Recommended Textbooks:

This course does not require any official textbook. A recommended free online textbook (with free, interactive text) can be found [here](#). More information is available on course website.

Software Requirements:

We will use Python (version 3.x) in this course. Download and install the latest version of Python from the [official website](#). Installation instructions are available in Tutorial-1 specification on the course website.

In addition, you would benefit from using an IDE (Integrated Development Environment). You are recommended to download and configure Visual Studio Code (VS Code) or the IDE of your choice. IDE installation instructions are available in Tutorial -1 specification.

- Download and setup VS Code
<https://code.visualstudio.com/docs>
- Python in Visual Studio Code
<https://code.visualstudio.com/docs/languages/python>
- Getting Started with Python in VS Code
<https://code.visualstudio.com/docs/python/python-tutorial>

Assessment Scheme:

In this course students will be evaluated according to the following criteria.

| Criteria | # | Total % | Tentative dates* |
|-------------------|--------------|---------|--|
| Practice Problems | Best 8 of 11 | 10% | weekly, beginning the week of September 13 th , due on Friday 11pm (see the course calendar) |
| Tutorials | Best 8 of 11 | 10% | weekly, beginning the week of September 13 th , due on Friday 11pm (see the course calendar) |
| Tests | 2 | 20% | October 21 and November 18 (more information will appear on Brightspace) |
| Assignments | 5 | 35% | biweekly, see the course calendar for due dates |
| Self Evaluations | 6 | 5% | associated with Assignments, more info will appear on Brightspace. |
| Final | 1 | 20% | scheduled by the Registrar |

*Dates are subject to change. Announcements will be made in lecture and on the course website.

Double pass rule:

The course uses a double pass rule. All assessment criteria are divided into two components; **Component-1:** practice problems, tutorials, assignments and self evaluation, and **Component-2:** tests and final exam. In order to pass the course, you **must pass both Component-1 and Component-2**. There will be no exceptions.

For example, if a student does not attempt tutorials, practice problems and self evaluation, but receives a 80% on each assignment, their assigned work receives a mark of $(0 * .01) + (0 * .01) + (0 * .05) + (80 * .35) = 28$ of a possible 60, and thus the student would receive an F in the course regardless of exam marks. The combined weight of Component-2 is 40% of your final grade. If you receive less than 20 out of 40 in this component, your final grade will be an F.

Practice Problems:

Every week a set of practice problems related to the lecture materials will be posted on Brightspace. These questions will test your knowledge on the concepts and grammar of the Python programming language. Hopefully, this will reinforce your understanding of these programming tools to think computationally and solve problems. Each problem set will open on Monday and is **due by the following Friday night (11 pm)**. Multiple attempts within the week are allowed. **Best 8 scores will be counted**. No late will be accepted.

Tutorials:

Attendance is mandatory. You must attend the tutorial session in which you are registered. You must submit your completed work on Brightspace at the **end of each tutorial week (Friday 11pm)**. In order to receive full marks, you must complete the work and demonstrate your understanding of the topic if asked by a TA. **Best 8 scores will be counted**. No late tutorials will be accepted.

Tests:

Tests are mandatory and will be conducted online. Tests will be held during the lecture time. You must attend, write, and submit your tests immediately upon completion in order to be graded. If you are unable to attend a test due to extenuating circumstances, you must inform the instructor via email **before the test begins**. You may miss **at most one (1)** test. There will be no make-up test but students who receive accommodations will have the weight of the missed test moved to the other. Accommodations are granted at the discretion of the instructor. Failure to follow the above instructions will result in a grade of 0 for your missed test.

Assignments:

Assignments are mandatory. All assignments will be made available in Brightspace and you will use Brightspace to submit your assignments. The assignment component of your final grade is computed from the score you receive on all five (5) assignments. **All assignments submissions must be your own individual and original work (see Plagiarism section below)**.

Assignment submission: multiple submissions are allowed before the due date. You are expected to work on your assignments consistently once they are released (uploading your progress periodically). As a result, **the instructor does not grant exemptions for the assignments due to sudden sickness, or any technical problems**, for example, problems regarding internet connectivity or computer hardware and/or software. **No provision is made for missed assignments, and no extra credit assignments will be available**. Therefore, you are advised to:

- periodically upload you progress (i.e., upload your progress at least daily).
- attempt to submit your final submission at least one hour in advance of the due date and time.

Late assignments are accepted for 24 hours after the posted deadline, but they incur penalties (more information will be available on assignment specifications). Assignment submissions are handled electronically, so there is no "grace period" with respect to the

deadline. 24 hours after the official submission deadline, you will not be able to submit your work.

For each assignment, you will be submitting one or more files that contain source code, and these files must be given the correct filename and be provided in the specified format. **Assignments that are incorrectly named or in the incorrect format will be penalized and may receive zero (0) mark.**

If any of the source code files you submit **does not run** it will receive zero mark. Consequently, after you upload your submission to Brightspace you must re-download it immediately and ensure that:

- your submission is a "zip" file that is not corrupt (i.e., it can be opened properly).
- each of your source code files can be run from an IDE or command line without error.
- each of your source code files can be viewed in a text editor (for marking purposes).
- your submission and each of your source code files follow the proper naming scheme.

You are expected to demonstrate good programming practices at all times, and your code may be penalized if it is poorly written.

Grading and Appeal:

All assignments, tutorials and tests submitted through Brightspace will be graded by the TAs. Practice problems will be auto graded. It is your responsibility to ensure that your marks (assignments, tutorials, tests, practice problems) published on Brightspace are correct within **ten (10) working days** of the date the marks were released. Concerns or complaints about the grading must be communicated first to the TA who marked your work, then, if the result is unsatisfactory, to the instructor within that time. After those ten days, all marks are considered final and will not be changed under any circumstances.

Please note that **a student cannot, for any reason, be exempted from more than 3 tutorials, more than 3 practice problems, or more than one test.**

Final Exam

Attendance is mandatory. The time and the format of the final exam will be announced later in the term. Registrar's office will schedule the exam time and more information can be found [here](#). The deferral process for formally scheduled exams is handled through the registrar's office as well, see the registrar's website for more details.

Bonus

Any assessment criteria that is not mandatory can be considered for bonus, e.g., any remaining tutorials and practice problems outside of your best submissions as explained above are available for bonus.

Bonus points are completely optional; not doing bonus points will not negatively impact your final grade. Bonus points cannot add more than 3% to your final mark, and it cannot move you from an F to a passing grade.

Learning Outcomes:

If a student engages with the course material and completes all assignments, tutorials and practice problems, then by the end of this course that student should be able to:

- Use a programming language to write computer programs (in the imperative paradigm)
- Explain the differences between algorithm design and implementation
- Apply different problem-solving heuristics (e.g., divide-and-conquer, abstraction)
- Explain the following topics:
 - data types, variable assignment, propositional logic, Boolean values, strings
 - branching, repeating, and nested control structures (i.e., "if", "for", "while")
 - data structures - lists, dictionaries, tuples etc.
 - functions, scopes, and recursion
 - objects and classes for data storage and manipulation
 - runtime analysis
- Implement some basic searching and sorting algorithms

Additional Notes

In addition to the time spent reading/viewing lecture materials and completing tutorials, students can expect to spend **at least ten (10) hours per week** on this course. Students are responsible for all course materials, including lecture notes, tutorial exercises, and all materials discussed in class and on any of the official discussion boards.

Students are asked to **pose all questions related to course content using the official discussion boards on Discord**; students **should not email the instructor directly** unless the question contains confidential information or is of a personal nature.

The instructor will attempt to answer every student email received within two (2) working days of the time the message was received, unless the email requests information already posted on Brightspace, Discord or in this course outline. To ensure that all announcements are received, **students are expected to check their email on a daily basis**.

All materials created for this course (including, but not limited to, lecture notes, recorded videos, in-class examples, tutorial exercises, assignments, examinations, and posted solutions) remain the intellectual property of the instructor. These materials are intended for the personal and non-transferable use of students registered in the current offering of the course. Reposting, reproducing, or redistributing any course materials, in part or in whole, without the written consent of the instructor, is strictly prohibited.

Plagiarism Policy:

Any student that violates academic integrity (intentionally or not) must be reported to the Associate Dean (Undergraduate) who will investigate the matter. The standard penalties for an academic integrity violation are as follows:

- First offence, first-year students (< 4.0 credits completed): No credit for assignment/activity in question, or a final grade reduction of one full letter grade (e.g., A- becomes B-, if reduction results in an F, so be it), whichever is greater.
- First offence (everyone else): F in the course
- Second offence: One-year suspension from program
- Third offence: Expulsion from the University

Note: these are minimum penalties. More-severe penalties will be applied in cases of egregious offences. Penalties for such offences can be found on the [ODS webpage](#).

There is a separate plagiarism policy document for this course that is located on Brightspace. Students must read this document thoroughly and must agree to adhere to this policy (and to all policies stated in this course outline) before the assignment resources will be made available.

If you are still unsure of the expectations regarding academic integrity (how to use and cite references, how much collaboration with lab or classmates is appropriate), you are invited to discuss any concerns with the instructor at the earliest opportunity.

Online Student Success Module:

This module has been designed to introduce the students to online learning and provide them with strategies and resources to help them successfully navigate this learning experience. It is brought to the students in collaboration with **Carleton's Centre for Student Academic Support (CSAS)** and **Teaching and Learning Services (TLS)**.

The module covers the main features, benefits and challenges of online learning and explores various tips, including how to effectively manage time, how to optimize the learning space, how to interact and engage online, and where to seek additional supports.

<https://carleton.ca/online/online-student-success-module/>

Undergraduate Academic Advisor

The Undergraduate Advisor for the School of Computer Science is available in Room 5302C HP; or by email at scs.ug.advisor@cunet.carleton.ca. The undergraduate advisors can assist with information about prerequisites and preclusions, course substitutions/equivalencies, understanding your academic audit and the remaining requirements for graduation. The undergraduate advisors will also refer students to appropriate resources such as the Science Student Success Centre, Learning Support Services and Writing Tutorial Services.

----- **University Policies** -----

For information about Carleton's academic year, including registration and withdrawal dates, see [Carleton's Academic Calendar](#).

Pregnancy 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, visit [Equity Services](#).

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, visit [Equity Services](#).

Academic Accommodations for Students with Disabilities If you have a documented disability requiring academic accommodations in this course, please contact the Paul Menton Centre for Students with Disabilities (PMC) at 613-520-6608 or pmc@carleton.ca for a formal evaluation or contact your PMC coordinator to send your instructor your Letter of Accommodation at the beginning of the term. You must also contact the PMC no later than two weeks before the first in-class scheduled test or exam requiring accommodation (if applicable). After requesting accommodation from PMC, meet with your instructor as soon as possible to ensure accommodation arrangements are made. For more details, visit the [Paul Menton Centre](#) website.

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 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: 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 [the policy](#).

Student Academic Integrity Policy. Every student should be familiar with the Carleton University student academic integrity policy. A student found in violation of academic integrity standards may be awarded penalties which range from a reprimand to receiving a grade of *F* in the course or even being expelled from the program or University. Examples of punishable offences include: plagiarism and unauthorized co-operation or collaboration. Information on this policy may be found [here](#).

Plagiarism. As defined by Senate, "plagiarism is presenting, whether intentional or not, the ideas, expression of ideas or work of others as one's own". Such reported offences will be reviewed by the office of the Dean of Science. Standard penalty guidelines can be found [here](#).

Unauthorized Co-operation or Collaboration. Senate policy states that "to ensure fairness and equity in assessment of term work, students shall not co-operate or collaborate in the completion of an academic assignment, in whole or in part, when the instructor has indicated that the assignment is to be completed on an individual basis". Please refer to the course outline statement or the instructor concerning this issue.