

# COMP SCI 1MD3 – Winter 2018

## Introduction to Programming

### Administrative Details

**Instructor:** Douglas Stebila <[stebilad@mcmaster.ca](mailto:stebilad@mcmaster.ca)>

**Office:** ITB 162

**TAs:**

- Victor Chen <[chenv5@mcmaster.ca](mailto:chenv5@mcmaster.ca)>
- Natalie Chin <[chinnh@mcmaster.ca](mailto:chinnh@mcmaster.ca)>
- Karl Knopf <[knopfka@mcmaster.ca](mailto:knopfka@mcmaster.ca)>
- Joey Legere <[legerejk@mcmaster.ca](mailto:legerejk@mcmaster.ca)>

**Website:** <http://avenue.mcmaster.ca>. A Facebook group is available for discussions, and the teaching team will answer questions posted there. Use of Facebook is not required and all official material will be posted to Avenue. Students are expected to behave appropriately on the Facebook group.

<https://www.facebook.com/groups/985102941628275/>

**Schedule:**

- **Lectures:** Tuesdays, Wednesdays, Fridays 3:30-4:20pm, ITB AB 102
- **Tutorials:**
  - T04: Mondays 10:30-11:20am, BSB 244 (Joey)
  - T01: Mondays 1:30-2:20pm, BSB 249 (Natalie)
  - T02: Mondays 4:30-5:20pm, BSB 249 (Victor)
  - T03: Tuesdays 2:30-3:20, BSB 249 (Karl)
- **Drop-in centre:** Daily, 9am-4:30pm, ITB 242 (John Nakamura)
  - (closed selected times including M, Th 12:30-1:30, Tu 1:30-2:30),
- **Office hours (Stebila):** Thursdays 2:30-3:30pm, ITB 162

If you would like to meet but cannot come during office hours, please email me to arrange a time.

**Prerequisites:** One of MATH 1K03, MATH 1LS3, Grade 12 Advanced Functions and Introductory Calculus U, Grade 12 Calculus and Vectors.

**Antirequisite:** ENG 1D04.

## Course Objectives

By the end of this course, students will have

- learned the essential parts of the Python programming language (sufficiently to learn more on Python on one's own)
- learned how to organize, write, document, test medium-sized programs,
- be aware of limits of computation,
- be able to bring an informal problem statement into a computational formulation,
- learned a number of algorithmic techniques for solving complex problems.

**Calendar description: 3 units.** Introduction to fundamental programming concepts: values and types, expressions and evaluation, control flow constructs and exceptions, recursion, input/output and file processing.

## Assessment

Meaningful and memorable	5%	due weekly on Mondays at 5pm
Labs	40%	due weekly on Wednesdays at 5pm
Test #1	15%	in class on Tue Feb 13, locations TBA
Test #2	15%	in class on Wed Mar 14, locations TBA
Final exam	25%	during final exam period

**Meaningful and memorable (M&M):** At the end of every week (starting the week of Jan. 8-14) you are asked to submit via Avenue Discussions a paragraph with what you learned in class – something “meaningful” and “memorable” from class. This allows you to reflect on the class and it gives the teaching team immediate feedback. Submissions will open after Friday’s lecture and are due the following Monday at 5pm. Once you make your submission, you can see what other students have written. Each submission counts 0.5%, to a maximum of 5%. As there are 12 weeks and 10 submissions, MSAFs or other reasons for absence do not count. Every sensible submission counts – if you think you haven’t learned anything, state why, and your submission still counts. Late submissions are not accepted.

**Labs:** The labs are organized around tutorials. You are given exercise questions in advance. You should attempt the exercises on your own, before the tutorial, for which you will need to consult a textbook and have access to Python 3 using Jupyter Notebooks. Completing these exercises is essential to succeeding in this course - you will not learn to program by simply trying to understand other people's programs! In case you get stuck, the TA running the lab can help you. You are allowed to collaborate with each other, but the code you submit must be your own. If you can complete the exercises on your own, you do not need to go to your tutorial, attendance is not checked. There are 11 labs and

each lab is worth 4%. We will drop your lowest mark, giving a total of 40%. Incorrect submissions will get partial marks. If you get an MSAF/faculty-approved relief, you receive a 3 calendar day extension. Late submissions are not accepted, since we drop your lowest mark. Some labs have bonus questions that count towards any lab: you can get more than 100% on one lab, but all labs in total are still at most 40% of the course grade.

**Tests:** There will be two midterm tests, each worth 15%. The midterm tests will be scheduled during regular classes on Tuesday February 13 and Wednesday March 14; locations to be announced on Avenue closer to the date. You will complete the test entirely on a computer, using either your own laptop or a university computer in a lab. If you get an MSAF/faculty-approved relief for one test, the weight will be added to the other test. If you get an MSAF/faculty-approved relief for both tests, there will be a separate test covering that material, which the instructor may conduct orally, written, or on a computer.

**Final exam:** Scheduled during the final exam period. You will complete the exam entirely on a computer, using either your own laptop or a university computer in a lab.

**Summary of extensions and late policy:**

- Meaningful & memorable: no MSAF/extensions, no late submissions, best 10 out of 12 count.
- Labs: MSAF => 3 calendar day extension, no late submissions, best 10 out of 11 count.
- Test #1 and #2:
  - 1 MSAF/faculty-approved relief => weight added to other test+exam;
  - 2 MSAFs/faculty-approved reliefs => separate test.
- Final exam: Standard university policy on missed exams applies.

For MSAFable items, you must submit the MSAF online at <http://www.mcmaster.ca/msaf/>.

Re-marking must be requested within 7 days of results being returned.

## **Outline of Topics**

- Introduction to Computer Science, Computer Systems, Python programming language, Computational thinking
- Values and types (innate data types, data encoding, expressions, variables, assignment, strings, lists, object & classes)
- Imperative programming (modules, flow of control, control structures -- loops, exceptions and exception processing, procedures and parameter passing)
- Input and output, files, and operations with files
- Data visualization
- Databases and machine learning

## **Tentative Timeline**

Week	Main topic	Lectures	M&M due	Tutorials	Lab due	Test
1	Course overview	Jan 5	None	Jan 8/9	None	
2	Programming & expressions	Jan 9-12	Jan 15	Jan 15/16	Jan 17	
3	Statements	Jan 16-19	Jan 22	Jan 22/23	Jan 24	
4	Functions	Jan 23-26	Jan 29	Jan 29/30	Jan 31	
5	Numerical computations	Jan 30-Feb 2	Feb 5	Feb 5/6	Feb 7	
6	Structured data types	Feb 6-9	Feb 12	Feb 12/13	Feb 14	Feb 13
7	Testing & exceptions	Feb 14-16	Feb 26	Feb 26/27	Feb 28	
Reading week (Feb 19-23)		None	None	None	None	
8	Classes	Feb 27-Mar 1	Mar 5	Mar 5/6	Mar 7	
9	Interaction and files	Mar 6-9	Mar 12	Mar 12/13	Mar 14	Mar 14
10	Data visualization	Mar 13, 16	Mar 19	Mar 19/20	Mar 21	
11	Databases	Mar 20-23	Mar 26	Mar 26/27	Mar 28	
12	Machine learning	Mar 27-30	Apr 2	Apr 2/3	Apr 4	
13	Advanced topics	Apr 3-6	Apr 9	None	None	
Finals						Exam period

The instructor and university reserve the right to modify elements of the course during the term. The university may change the dates and deadlines for any or all courses in extreme circumstances. If either type of modification becomes necessary, reasonable notice and communication with the students will be given with explanation and the opportunity to comment on changes. It is the responsibility of the student to check their McMaster email and course websites weekly during the term and to note any changes.

## **Textbook**

The course does not follow strictly a single textbook, but you are advised to get a textbook, in addition to using the language and library reference at [python.org](http://python.org). The most recommendable textbook is:

- John V. Guttag, [\*Introduction to Computation and Programming Using Python with Application to Understanding Data, second edition\*](#), MIT Press, 2016. ISBN 9780262529624

The book will be available from the Campus Store, and is also available for purchase on Amazon and Chapters/Indigo, in both paperback and eBook formats.

Many online resources on Python programming exist, but in general they are not thorough enough to be a substitute for a proper textbook. Another recommendable textbook includes:

- John M. Zelle, [\*Python Programming: An Introduction to Computer Science—2nd Edition\*](#) Franklin, Beedle, and Associates, 2010.

Whatever resource you use, ensure that it uses **Python 3**, not Python 2. In particular, older editions of both books above were only about Python 2.

This semester I will be using iClickers for polling during class, so if you have an iClicker, please bring it. There will be no marks associated with the iClicker responses, so it is entirely optional to purchase one. The basic iClicker+ (with just the buttons, no display) is fine. If you have a mobile phone, you can also use the iClicker Reef app, but it apparently requires an in-app purchase.

## **Getting Assistance**

- The primary means for getting help are talking to the TAs during the tutorials. The best is to get together in a group and then ask the TA.
- You can consult one of the instructional assistants of the departmental Drop-In Centre, see for times above. The Drop-In Centre offers assistance to all Level 1 students taking a course by the Department of Computing and Software.
- The Facebook discussion group (link above) will be monitored by the teaching team. You can generally expect a quick reply there, within 24 hours from the teaching team, and possibly sooner from your fellow students. While collaboration is permitted and encouraged, don't post any solutions, and please be courteous to everyone on the group.
- You can also ask questions by e-mail. You can generally expect a reply within 24 hours, but not necessarily on the same day. If a lab is due or you are preparing for a test, don't expect e-mail replies within minutes!

## **Academic Integrity**

You are expected to exhibit honesty and use ethical behaviour in all aspects of the learning process. Academic credentials you earn are rooted in principles of honesty and academic integrity.

Academic dishonesty is to knowingly act or fail to act in a way that results or could result in unearned academic credit or advantage. This behaviour can result in serious consequences, e.g. the grade of zero on an assignment, loss of credit with a notation on the transcript (notation reads: "Grade of F assigned for academic dishonesty"), and/or suspension or expulsion from the university.

It is your responsibility to understand what constitutes academic dishonesty. For information on the various types of academic dishonesty please refer to the Academic Integrity Policy, located at <http://www.mcmaster.ca/academicintegrity>

The following illustrates only three forms of academic dishonesty:

1. Plagiarism, e.g. the submission of work that is not one's own or for which other credit has been obtained.
2. Improper collaboration in group work.

3. Copying or using unauthorized aids in tests and examinations.

### **Academic Accommodation for Students with Disabilities**

Students who require academic accommodation must contact Student Accessibility Services (SAS) to make arrangements with a Program Coordinator. Academic accommodations must be arranged for each term of study. Student Accessibility Services can be contacted by phone 905-525-9140 ext. 28652 or e-mail [sas@mcmaster.ca](mailto:sas@mcmaster.ca). For further information, consult McMaster University's Policy for Academic Accommodation of Students with Disabilities.