# intro to CS

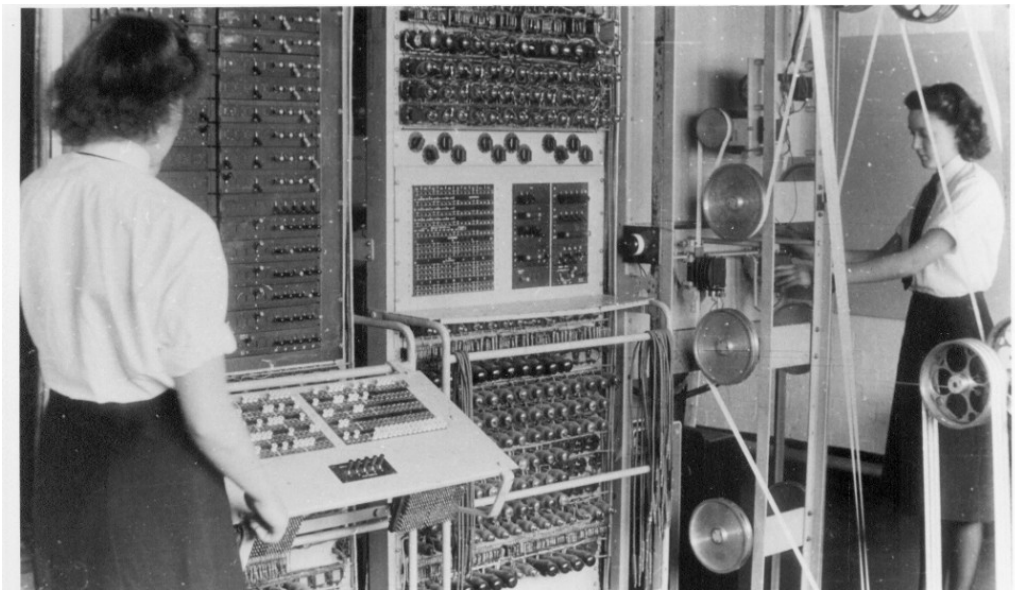
COMP 1631

FALL 2018

MWF 8:30AM
or
MWF 1:30PM

CRABTREE M14
or
DUNN 113

MOUNT ALLISON UNIVERSITY


CROWN COPYRIGHT, USED WITH KIND PERMISSION, DIRECTOR GCHQ

## Course Description and Objectives

In this course, we explore fundamental computational thinking strategies to solve problems. No prior experience in computer science is required.

You will develop computational problem-solving skills by programming in the Python language. You will explore broad themes that arise in the study of computer science. Most importantly, you will discover that computer science is much more than "just" programming and that computing-related studies are relevant for all.

Computing is much larger than just computer science: computational thinking and data literacy are transforming the way we live and work. The convergence of physical sciences and life sciences with computing have produced significant advances in how we understand the world and ourselves, e.g., the human genome project.

Digital humanities uses computational thinking to analyze literature, making connections that were more challenging or even impossible without data analytic techniques.

The digital currency revolution represents the future of money and global finance. Other examples of the impact of computing include the ability to track deforestation in the Amazon, track the effects of climate change, track global flu trends and make critical information more accessible in a disaster.

### Syllabus Contents


| | |
|---|---|
| Course description | 1 |
| Required texts | 1 |
| Course requirements | 2 |
| Policies and procedures | 3 |
| Course schedule | 4 |


### Course details

Website: https://moodle.mta.ca

Required online texts:

- *CS for All* by Christine Alvarado, Zach Dodds, Geoff Kuenning, and Ran Libeskind-Hadas
- *How to think like a Computer Scientist* (Python 3 edition) https://runestone.academy

## How to take this course

Not everyone comes to this course from the same place or has the same ambitions. Some of you are here for a 10 week science credit, while others hope to gain something more. Before we begin, give some thought to what kind of student you are, and why you are here.

It is entirely possible to do well in this class without being transformed by your newfound computer science knowledge, but it would be a darn shame. We like to think that this course operates on three levels, so indulge us in a metaphor. We're on a shore, and this course is the ocean. How far out you go is entirely up to you...

## wading

You need the basic ideas of computational thinking, the highlights, the main ideas, the surface-level knowledge.

There's nothing wrong with staying in the shallow; this approach may work for you if this is likely to be your only computer science course, or if you've never taken one before and it's all new.

"Waders" will tend to assume that the textbooks and professor are mutually reinforcing, telling basically the same story. Waders are mainly concerned with WHAT must be done to solve a problem.

## snorkeling

You have a grasp of the basics and are ready to think computationally and explore what's below the surface.

Perhaps you've taken computer science before, or you're a beginning computer science major. You already know that computer science is not just programming and involves considering the efficiency of a solution, not just that a solution exists.

"Snorkelers" notice that there are different ways to solve a problem and they respectfully challenge assumptions through lively debate. Snorkelers are interested in HOW and WHY a solution is reached.

## scuba diving

You want to go deeper into the computational thinking process, using critical thinking and more advanced problem-solving strategies to determine how a solution to an extension of the problem might be solved.

Experienced? You are well aware of computational problem-solving strategies and how algorithms are constructed. You actively seek strategies that lend themselves to a more general setting.

"Divers" don't take any of the course's structure or content as natural or inevitable. They see (and then fill) the course's gaps. They are curious, passionate, and concerned with WHY COMPUTER SCIENCE MATTERS.

## Course Requirements

**Class Participation.** We do not take attendance but do regularly complete in-class activities. You will also be responsible for completing pre-class quizzes on Moodle. You should arrive on time and be prepared for lecture and discussion. There is no such thing as an "excused" absence; you're either in class or not.

**Exams.** You will have three exams in this course: an in-class practice midterm worth 1% on Oct. 5, a midterm exam worth 20% on Oct 15 (from 7PM - 8:30PM) and a cumulative final exam worth 40% (date to be determined). Exams consist of multiple choice and problem-solving questions similar to the labs and in-class activities.

**Labs.** Labs take place in Dunn 102 and begin the week of Sept. 10. Please attend the lab section for which you are registered. Lab attendance is mandatory. *You will lose 2% for each unexcused absence.* Mr. Rob Allen (rallen@mta.ca) should be the primary contact for any concerns regarding lab attendance.

Admiral Grace Hopper created the first compiler and coined the word "bug" to refer to an a error in a computer program.

### Marking Scheme

| | |
|---|---|
| Assignments | 25% |
| Exams* | 61% |
| Moodle quizzes | 4% |
| Labs | 10% |

*Average of under 50% on Exams component will result in an automatic failure of the course.

### COMP 1631 Teaching Team

| | | |
|---|---|---|
| Mr. R. Allen | Dunn 103 | rallen@mta.ca |
| Dr. M. Betti | Dunn 209 | mbetti@mta.ca |
| Dr. L. Ricker | Dunn 301 | lricker@mta.ca |

## Help & Resources

### If you are feeling lost or overwhelmed…

1. Make an appointment with us

You can email any member of the COMP 1631 teaching team and make an appointment us during our normal office hours or another convenient time. Many questions and issues can be resolved this way.

2. Find the Math Resource Centre

The student-staffed centre (currently located in the Bell Library) is available for computer science help too! Watch Moodle for further announcements

3. Contact the Wellness Centre

Mount Allison course instructors and student services cooperate to assist in helping students manage course and life stressors. Student services and the wellness centre can both be found in the Wallace McCain Student Centre. Additionally, help available through a counsellor can be reached at counsellor@mta.ca and www.mta.ca/wellness

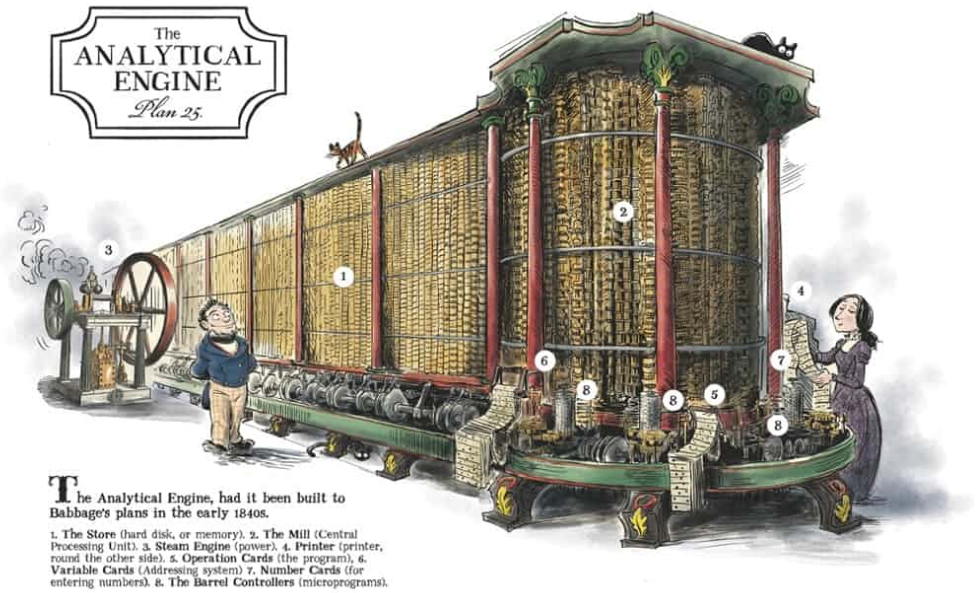
4. Work with your classmates

The data is in and the results are clear: students who study with peers perform better in courses, and show greater understanding of the material. Find some classmates and start reading and talking together.

## Accommodations

Accommodations for documented disabilities are arranged through registration with the Meighen Centre, located on the third floor of the Wallace McCain Student Centre.

http://sydneypadua.com/2dgoggles/the-marvellous-analytical-engine-how-it-works/




The Analytical Engine, had it been built to Babbage's plans in the early 1840s.

1. **The Store** (hard disk, or memory). 2. **The Mill** (Central Processing Unit). 3. **Steam Engine** (power). 4. **Printer** (printer, round the other side). 5. **Operation Cards** (the program). 6. **Variable Cards** (Addressing system) 7. **Number Cards** (for entering numbers). 8. **The Barrel Controllers** (microprograms).

# Course Policies

**On Plagiarism.** All forms of academic misconduct are dealt with seriously at Mount Allison University. It is your responsibility to read and be familiar with the academic regulations of the University as presented in the Calendar. Be sure that your understanding of the concepts as described in *Section 10.6: Academic Integrity* is clear, and that you understand your responsibilities in preparing and representing your work.

In any assignment, you must cite the sources of your information, code, quotations, or ideas that are not your own. By handing in any work for this course, unless you have specifically identified any other authorship, you are claiming that the sole authorship is your own. Unless otherwise indicated, students are expected to have completed all assignment and in-class activities as independent efforts.

We regularly use software to detect any unauthorized collaborations. If any such collaborations are found, grade sanctions will be applied, and the matter will be referred to the Registrar's office as per the regulations described above.

**On Technology in class.** There is convincing evidence that electronic distractions –text messages, email, Facebook, etc.– interfere with your ability to focus during lectures. Silence your phones and put them away. Do not bring laptops to class unless you have a documented disability that requires their use. If a specific, coding-based activity is to be held in class, your instructors will indicate that on that special occasion, you are encouraged to bring your computer to use in class. Otherwise the above policy will be in effect.

**On Assignment deadlines.** Assignments are due before 11:55pm on the day indicated on the course Moodle page. If you are unable to meet the deadline or want to demonstrate additional understanding of the material, you have a chance to resubmit an assignment within one week of receiving feedback. In this case, the resulting mark will be an average of the two submissions.

Consult a member of the teaching team if prolonged illness will affect your ability to meet course expectations.

| Week | Topic | Reading |
|---|---|---|
| Sept 8 | Python and Picobot | *CS for All* Section 1.1;<br>*How to think like a Computer Scientist* Chapter 1 |
| Sept. 15 | Functions | *CS for All* Section 1.2;<br>*How to think like a Computer Scientist* Chapter 2 |
| Sept. 22 | Functions & Recursion | *CS for All* Chapter 2;<br>*How to think like a Computer Scientist* Chapter 15 |
| Sept. 29 | Recursion | *CS for All* Chapter 2;<br>*How to think like a Computer Scientist* Chapter 15 |
| Oct. 6 | Algorithm design | *No new reading* |
| Oct. 13 | Thinking in binary | *CS for All* Chapter 4.1 - 4.2; |
| Oct. 20 | Binary circuits | *CS for All* Chapter 4.3 - 4.4; |
| Oct. 27 | Avengers assembly? | *CS for All* Chapter 4.4 - 4.5; |
| Nov. 3 | Loops | *CS for All* Chapter 5;<br>*How to think like a Computer Scientist* Chapter 8 |
| Nov. 17 | Baby's first 2D data structure | *CS for All* Chapter 7; |
| Nov. 24 | Finite state machines | *No new reading* |

**Dates of interest**

| | |
|---|---|
| Sept 14 | Last day to register for Fall term 3 credit courses |
| Sept 28 | Last day to drop Fall term 3 credit courses (changes after this date will result in a W recorded on your transcript) |
| Oct. 8 | Thanksgiving (no classes) |
| Oct. 15 | Midterm (7pm - 8:30 pm) |
| Nov. 12-16 | Fall Break |
| Dec. 4 | Last day of MWF classes (make up day for Thanksgiving) |