COMP 102: Computers and Computing

Overview

This course assumes no previous computing or college-level math experience. The goal is to survey many of the concepts and technologies that make computers and computing so fundamental to our modern world. Just as understanding English literature requires understanding the English language, understanding computers and computing requires some understanding of programming. As a result, learning the fundamentals of programming will be a cornerstone of this class – and a major objective of the semester. Through the course, students will develop familiarity with bash, SQL, and python. Through them, we will explore several of the key applications of computing in our everyday lives: data mining, web services, cryptography, and automation.

Do I need to know how to program?

NO! You will see "programming" mentioned in many places throughout this syllabus. That's because this is a skill you will learn in this class. The only thing we assume you know how to do is load a web browser. We'll work together to learn the rest.

Class Schedule

Room: Trottier 1090
Time: 2:35 PM – 3:55 PM (Tuesday/Thursday)

Contact Information

Instructor: Professor Derek Ruths
Email: derek.ruths@mcgill.ca
Phone: 514-398-7079
Office: Trottier 3105
Office Hours: Tuesday 1 – 2:30 PM @ Trottier 3105

Teaching Assistants:

Haji Mohammed Saleem (haji.saleem@mail.mcgill.ca)
Office Hours: Trottier 3110, TBD
Koustuv Sinha (koustuv.sinha@mail.mcgill.ca)
Office Hours: Trottier 3110, TBD

Course Structure

We'll be using an inverted class structure. This means our "lectures" happen outside of class, and we do discussion and practice problems during class. There are several important implications of this approach.

1. You have to keep up with the posted materials. You need to take responsibility for this yourself – make protected time to do the readings/watch the videos and attempt the practice problems before class.
2. I want you to come to class, but you don't *have to* come to class. Homeworks will be based around materials that you'll read on your own time. We use class time to flush these concepts out and deepen our understanding of them. Think of classes as tutorial sessions.

The content for this course is broken into modules, each lasting (about) a week. For a given module, our schedule will work like this:

- The week before we start the module in class
    - Thursday night I will post on MyCourses:
        - Links to learning materials for following week
        - Practice problems we'll discuss in class the following week
        - Homework, which covers the module we have just finished
- The week of the module
    - Before Tuesday class,
        - have viewed/read all linked materials
        - have attempted the practice problems
    - Tuesday and Thursday in class we will field questions about the material and work together through the practice problems (and other problems that seem relevant).
    - Thursday night, the homework from the previous week is due.

If all this seems complicated – please don't worry. It's just clunky to explain in words – in practice, it's very simple: prepare for class before Tuesday; Tuesday – Thursday attend class and we'll puzzle through fun/tricky problems; Thursday submit your homework assignment.

Textbook

There is no designated textbook for this class. However, there will be weekly reading and viewing assignments for which material/links will be provided.

Coding Tools (You'll learn how to use these!)

Nearly all activities in and assignments for this class will involve using the computer and writing code. For this, we'll be using the codenvy.io platform. Each student will need to create their own developer account (it's free). Don't worry if you're not familiar with codenvy (or anything about programming): the first module will walk you through how to set it up and use it.

Grading

Each student's final grade in this course will be determined by approximately 12 weekly assignments and a final exam. The grade breakdown will be:

75% assignments
25% final exam

Assignments. Assignments will be assigned on Thursday, due the following Thursday at midnight. Typically, they will be programming assignments. If you haven't programmed before, don't panic. We'll start out very gradually – again, we don't assume that you arrive on the first day knowing anything more than how to launch a web browser.

Final Exam. The final exam will test understanding of core concepts covered throughout the course. The emphasis of this exam won't be on writing code, but rather demonstrating an understanding of how computation works in the modern world and how such systems are built from the pieces we've seen over the course of the class.

Late Assignments
If submitting an assignment by the due date presents a problem, contact me as soon as possible to determine whether a late submission can be accommodated. If a later due date has not been arranged, then the late assignment's final grade will be penalized at 10% per day.

Extenuating Circumstances
I want every student in this course to succeed. If unforeseen situations arise that interfere with your ability to complete coursework or devote adequate time to this course, *please contact me as soon as you suspect there could be a problem.* While I cannot guarantee that I will oblige every request and situation, the sooner you notify me of the situation, the sooner we can work to find a way to accommodate any issues you may be dealing with. Please bear in mind that **requests that have waited till the last minute will not be accommodated.**

Most crucially – software not working is NOT considered extenuating circumstances. It's important that you start homework early enough to handle some of the bumps that naturally come along with programming. We're going to do some awesome stuff this semester – don't save all your work till the night before it's due. You probably won't get the work done and, even if you do, you won't learn much.

Academic Integrity

Except where specifically noted, homework may be discussed with other students and I encourage group work. However, all work (code, writing, and answers) must be the student's own. Copying another student's work, in any form, constitutes an act of cheating. Here's a rule of thumb:

- BAD: Looking at someone's work because you don't know how to do it (or haven't done it) yet.

- GOOD: Looking at someone's work because you know how to do it and are trying to give them some helpful direction.

The final will both be administered as in-class (subject to change depending on a classroom large enough for us), closed book exams.

McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.mcgill.ca/integrity for more information).

Right to Submit Work in English or French
In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Schedule

Lecture 0: What is computing?

Lecture 1: Internet, the cloud, web services, and codenvy

Lecture 2: Debugging and the UNIX command line

Lecture 3: Algorithms, variables, and bash scripts

Lecture 4: Data mining and regular expressions

Lecture 5: Web services, REST-ful APIs, and JSON

Lecture 6: Database Design

Lecture 7: Simple database queries

Lecture 8: Complex database queries and data mining

Lecture 9: Application architecture and Python programming

Lecture 10: Building a web app using Python

Lecture 11: Cryptography