CMPUT 274 Fa18 - INTRO TO TANGIBLE COMPUT I Combined LBL Fa18

Course Outline (2018)

CMPUT 274: Tangible Computing I -- Course Outline

General Information

Term: Fall 2018
Location: ETLC 2-005/2-009. Both numbers refer to the same, combined room.
Times: Tuesday and Thursday, 8:00 AM - 10:50 AM. Morning Problems start at 8 AM sharp.

Professors:

Paul Lu (paullu@ualberta.ca)

Ron Kube (rkube@ualberta.ca)

Contact Policy:

In general, posting to the eClass forum will get you the best and fastest response. For private matters, contact the Professors (see above).

If there is no personal or private content in your email, we may ask you to repost to the eClass forum. That way, everyone in the class gets the benefit of the same information.

Teaching Assistants:

Teaching Assistants:

Jason Cannon <jmcannon@ualberta.ca>
Patrisha De Boon <deboon@ualberta.ca>
Negar Hassanpour <hassanpo@ualberta.ca>
Taher Jafferjee <jafferje@ualberta.ca>
Arseniy Kouzmenkov <kouzmenk@ualberta.ca>
Joseph Meleshko <jmeleshk@ualberta.ca>
Daniel Mitchell <daniel7@ualberta.ca>
Veronica Salm <vsalm@ualberta.ca>
Sean Scheideman <searn@ualberta.ca>
Everton Schumacker Soares <schumack@ualberta.ca>
Yunpeng Tang <yunpeng5@ualberta.ca>
Noah Weninger <nweninge@ualberta.ca>

Office Hours:

All office hours will be starting **Wednesday, September 5**. Location is CSC 3-23 (the Arduino Lab in Computing Sciences Centre).

Lab Availability (ETLC 2-005/2-009):

A schedule should be posted on the door. At times outside of class, only half of the lab is available. Be aware that there are a variety of evening and weekend users of the lab. Of course, the lab is also unavailable on holidays when the university is closed. Advice: Set up and use the virtual machine (VM) on your own laptop.

Course Policies

CMPUT 274 is subject to the Department of Computing Science Policies. In particular, pay attention to these Computing Science Course Policies especially, as they relate to collaboration.

Overview

CMPUT 274/275 is a 2-course introduction to computer science. It integrates the introductory computing courses (CMPUT 174/175) with the second-year introductory algorithms course (CMPUT 204) and the second year introductory systems course (CMPUT 201). The course uses a problem-based approach to motivate the concepts and illustrate their application. It will be using the Arduino concrete-computing platform so that students will both become familiar with the typical screen-keyboard-mouse style of computing, but also the kind of embedded computing that is behind the scenes in the many devices that surround us. Delivery is hands on, with the classes taking place in a

science in the many devices that surround us. Delivery is hands-on, with the classes taking place in a lab environment.

For 2018, we are shifting the order in which material will be presented in CMPUT 274/275. The first language will be Python, followed by Arduino and C++.

Objectives

At the end of this course sequence, students will have covered sufficient content to satisfy the CMPUT 174, 175, and 201 pre-requisites of more advanced computing courses. In a change from previous years, you will be allowed to take CMPUT 204 for credit after taking 274/275. In fact, some senior courses (e.g., CMPUT 304) require CMPUT 204 explicitly. Other courses (e.g., CMPUT 379) will continue to accept either CMPUT 204 or 275 as a pre-requisite. Check with the Calendar and instructors for details on pre-requisites.

Pre-requisites

Basic high-school mathematics and comfort using a computer. No prior programming experience is necessary, but having prior programming experience is definitely a plus.

Students considering taking the course who have no or very limited previous programming experience can expect to spend a **significant extra time** on practice problems at home. The **fast pace environment** of the class means that concepts, ideas, techniques are introduced once and may not be repeated at all later, yet mastering these is necessary to keep up with the class: Students taking this class are expected to take responsibility for their learning, meaning that if you miss something in the class, you have to work on filling the gap on your own. Of course, the TAs are available to help in their office hours (and also the instructors to a lesser extent as in this class the instructors have no office hours), but if you are not asking for help, you may find it later that you are unable to follow the material.

Course Topics (274/275)

This course is structured to present fundamental computing concepts with a holistic problem-driven approach. In other words, between CMPUT 274 (and 275) you will learn these topics but they will "sneak up on you". Concepts and topics will be introduced *on demand* as they become necessary to solve real problems.

- Computing science concepts
  - Introduction to machine learning
  - Computational thinking

- Programming Concepts
  - Basics: variables, statements, conditionals, loops
  - Functions
  - Recursion
  - Arrays
  - Data structures
  - Testing and debugging
- Algorithm Design and Analysis
  - Problem solving: brute force, divide and conquer, graph algorithms, dynamic programming
  - Abstracting problems
  - Big-O notation
- System Concepts
  - Hardware basics: circuit design, input, output
  - Programming pipeline: compile, link, execute
  - Makefiles
  - Memory hierarchy
  - Caching

Course Work and Evaluation

Course Work	Schedule/Due Dates	Weight
Morning Problems	Twice a week in class, starting in the second or third week of classes	10%
Exercises	Mondays, 11:55pm, submitted using eClass	15%
Assignment (2 assignments; possibly in parts)	Dates TBA.	25%
Project	Nov 8 (proposals), Last 2 days of classes in December (demo)	20%
Final Exam	See University schedule	30%

Grades are unofficial until approved by the Department and/or Faculty offering the course.

Lectures

In this course we make no distinction between lecture and lab. Since we are always meeting in a lab environment, there will be both teaching and doing activites in every class. Thus, it is important to

attend all lectures.

Morning Problems

(Collaboration Policy = Solo Effort) Starting in the second or third week of classes, in the first 30 minutes of each class you will solve "small-scale" programming problems to reinforce good programming habits (i.e., the ability to write small code without much help -- a key skill for any programmer). For half marks, you can submit the solution online even after the in-class deadline within one week.

These programs are graded automatically and you receive either full marks by submitting a working solution in the class time, half marks by submitting a working solution within one week, or no marks if you either do not submit a solution or your solution does not pass the grading system.

These problems must be solved individually. It is important to emphasize that in the past, among the marks that could be earned on the various components, the marks earned on the morning problems were the **best predictor of the mark on the final**, as well as the overall mark in the class.

Exercises

(Collaboration Policy = Solo Effort) An exercise will be posted to be completed and submitted online by the next Monday at 11:55pm. Usually, exercises are released on a Wednesday. Unlike the assignments and projects and just like the morning problems, **the exercises are to be completed individually** without consultation with anyone besides instructors and TAs.

Assignments and Projects

(Collaboration Policy = Teamwork, where team size = 2, unless authorized by professors) There are two major assignments (possibly with multiple parts) and a project. The difference between an assignment and project is that you invent and work on your own project idea (subject to instructor approval). They will typically involve milestones announced in class that must be completed prior to the final deadline. We will be working on aspects of the assignments in class. In fact, the assignment will be used to motivate the material we examine in class: this is the key idea in a problem-based approach to learning. **You will be able to collaborate with exactly one other student on the assignments and the project.**

The assignments and the project must be completed in pairs: each student must pair up with precisely one other student. Students may not complete these alone or in a team with more than two students.

Late Policy

No late assignments will be accepted. Medical excuses will be considered.

Missed Assignments

A student who cannot complete a term assignment due to incapacitating illness, severe domestic affliction or other compelling reasons should contact the instructors as soon as possible. Situations will be handled on a case-by-case basis.

Deferred Final Examination

A student who cannot write the final examination due to incapacitating illness, severe domestic affliction or other compelling reasons canapplyfor a deferred final examination. Such an application must be made to the student's Faculty office within two working days of the missed examination and must be supported by a Statutory Declaration or other appropriate documentation (Calendar section 23.5.6).

Deferral of an assignment or final exam is a privilege and not a right; there is no guarantee that a deferral will be granted. Misrepresentation of Facts to gain a deferral is a serious breach of the *Code of Student Behaviour*.

Course Materials

There is no textbook for this course. However, a variety of resources will provided via hyperlinks from eClass.

There is a closely related massive open online course (MOOC) (Private Session, use your CCID: https://coursera.org/groups/problem-solving-programming-video-games-imvdg/invitation), called "Problem Solving, Programming and Video Games (PVG)". Material from the MOOC may be tested on the Final Exam.

Each student in CMPUT 274/275 will have to purchase the Arduino computer kit that we will be using in the course. All other resources will be available on the web. Each kit will be $200 or less (final price to be decided), and is used for both CMPUT 274 and 275. So the cost of the kit is less than typical textbooks for two courses. Kits will be available in October.

The development software for the Arduino runs on Mac OS X, Windows (XP or 7+), and Linux

(Ubuntu for sure), so it should work on any of your personal machines. We will be using a virtual machine in the labs, which will be made available if you want to use the same environment on your own personal machines. **We will only support the virtual machine environment**.

Academic Integrity

The University of Alberta is committed to the highest standards of academic integrity and honesty. Students are expected to be familiar with these standards regarding academic honesty and to uphold the policies of the University in this respect. Students are particularly urged to familiarize themselves with the provisions of the Code of Student Behaviour (online here) and avoid any behaviour which could potentially result in suspicions of cheating, plagiarism, misrepresentation of facts and/or participation in an offence. Academic dishonesty is a serious offence and can result in suspension or expulsion from the University. (GFC 29 SEP 2003)

For a more detailed description of plagiarism, cheating, and misrepresentation of the facts, consult the university's "Don't Cheatsheet".

Student Support Services

STUDENTS ELIGIBLE FOR ACCESSIBILITY-RELATED ACCOMMODATIONS (students registered with Student Accessibility Service - SAS)
Eligible students have both rights and responsibilities with regard to accessibility-related accommodations. Consequently, scheduling exam accommodations in accordance with SSDS deadlines and procedures is essential. Please note adherence to procedures and deadlines is required for U of A to provide accommodations. Contact SSDS (www.ssds.ualberta.ca) for further information.

STUDENT SUCCESS CENTRE

Students who require additional help in developing strategies for better time management, study skills or examination skills should contact the Student Success Centre (2-300 Students' Union Building).

Last modified: Tuesday, 4 September 2018, 1:03 PM

◀ Discussion Forum Jump to... Course Outline (2017)

(hidden) ▶

Moodle Docs for this page

You are logged in as **Paul Lu** (**Log out**)

CMPUT 274 (Fall 2018 LBL A1 EA1)