CSC491/2600

University of Toronto's Capstone Design Course through the Department of Computer Science Innovation Lab (DCSIL).

Website Table of Contents

Course Overview

Students will write a software application that implements the business ideas developed in CSC454/2526 (Business of Software). This course will expose students to the core technologies, ideas, and processes in developing a startup.

Students will be given a theme, in the corequisite course (CSC454), in which to develop a startup. Students are expected to:

- Define their own scope of problem within the theme
- Develop a cohesive plan
- Produce a working MVP (minimum viable product)
- Present their work in various mediums, including written, orally, visually, and through the internet.

Class time will be a mixture of project-focused workshops, lectures, and discussions. The class will be small and very interactive. Students may hear from guest lecturers from the field.

Requirements to take this course

- There is a co-requisite to CSC491/2600, students must also be enrolled in CSC454/2527 or have taken CSC454/2527 before to take this course
- Students must maintain a good GPA
- Students must have some prior software experience (web applictions preferred, though not required)
  - I would also encourage you to review the recommended skills and knowledge on the DCSIL website
- Students must have at least 4 CSC300+ level courses, and any other 300+ level course (previously 5 CSC300+ level courses)

Important: There is also an application process on the DCSIL website.

Expectations and evaluation of Students

In general we expect students to come prepared to be guided, mentored, and to work hard throughout the term. This is not an easy course and there is a lot of work, but it is a course former students regularly describe as "having taught me more than all my other courses" and "having taught concepts I use on a daily basis as a software engineer". Despite this, the teaching staff make themselves available via Slack and Zoom as needed and will help guide and mentor you. The teaching staff is here to help you succeed as long as you put the effort in, effectively contribute to your team, and communicate (that last one is important!) In order for students to be successful in this course and their team, here is a subset of expectations students should have:

- Students will actively participate in all assignments and communicate regularly with their team
  - Any absences will be communicated in advance to both the team and instructor
- Students will write code and substantially contribute to their team's software/data stack.
  - Doing product or design work is great, but it is not a replacement for software in this course. Of course, if you are also designing or doing another role, we won't expect *as much* coding. Talk to the instructor to make sure you're striking a good balance.
- Students will adapt and self-teach any technologies their team chooses for their team's project (Instructor does not choose the techs)
- Students will record all work using GitHub Issues, GitHub Pull Requests, and learn the GitHub / Git platforms if required

All of these expectations are in place, combined with mentorship and guidance, so that we can evaluate your abilities around:

- Write software and learn new technologies as required
- Design and architect software systems
- Manage and execute software projects
- Collaborate with other roles (business, finance, etc)
- Set up a proper "Software Development Lifecyle" (SDLC) including continuous integration, testing, and deploys

With these expectations and evaluation principles, we know you'll be better prepared for the software industry.

Required Materials/Software

There aren't any *required* materials *per se*, but it is required that you bring a laptop to class with a fully functioning developer environment for your project. iPads/tablets are not recommended as the main vehicle unless you can develop on them.

You will also need a GitHub account.

It is **required** to download, install, and set up Zoom as this will be used for all communications throughout the term.

Lecturer

This course is being updated. The lecturer will be announced as the course is offered again.

Office Hours

TBD.

Time and Location

Class is on Wednesday from 6-9PM EST. If you are not in a time zone that aligns with EST, please let me know early!

- If we are in person, we will meet in the DCSIL Lab in Gerstein Library
- If we are online, we will meet on Zoom

We use primarily an inverted classroom. This means that lectures are pre-recorded and asynchronous online. The 6-9pm lecture slot is reserved for working periods, guest lectures, discussion, etc. Students are expected to attend class times in order to work with their team, attend valuable tutorials, demo their work, and work with the instructor.

I don't like something about this course

There's always room for improvement and I'm happy to take any feedback.

Please submit an issue on this repo, or preferably make a pull request changing or adding something you don't like.

That said, remember that your changes or concerns aren't guaranteed to be addressed how you want. I (the instructor) have the final say.

If you prefer to speak privately, please email me or message me on Slack.

Class Schedule

This is a class schedule indicating what we aim to do each week and the recommended lectures to watch each week. This is broken down by weeks and aims to provide you a general outline of the progress you should make throughout this course.

NOTE: Lecture Time is subject to change with the needs of the course, industry partners, instructor, TA, and students. You should always bring a laptop and be ready to work on your project and with your team.

#	Date	Assignments	Lecture Time	Recommended Async Lectures
1	Wednesday, January 11 2023	-	First day of class, Introductions & Course Overview	Story Telling
-	Monday, January 16 2023	❗A1 Due	-	-
2	Wednesday, January 18 2023	-	TBD	Various Diversity & Inclusion
-	Monday, January 23 2023	❗A2 Due	Instructor away, TA Tutorial on Cloud Technologies	Various Diversity & Inclusion
3	Wednesday, January 25 2023	-	TBD	Different Kinds of Tests / Best Practices - Languages / Best Practices - Linting, Semantic Analysis, etc
4	Wednesday, February 1 2023	-	TBD	Different Kinds of Tests / Best Practices - Languages / Best Practices - Linting, Semantic Analysis, etc
-	Monday, February 6 2023	❗A3 Due	-	-
5	Wednesday, February 8 2023	❗A6 Demo (1/4)	CI & Testing Lecture	UX Research + Data Bias
6	Wednesday, February 15	-	Work in Class	Ethics and Accessibility

#	Date	Assignments	Lecture Time	Recommended Async Lectures
	2023			
-	Sunday, February 19 2023	❗A4 Due	-	-
7	Wednesday, February 22 2023	-	Reading Week	-
8	Wednesday, March 1 2023	-	TBD Tutorial	Infrastructure, Prod Eng, Production, etc
-	Monday, March 6 2023	❗A5 Due	-	-
9	Wednesday, March 8 2023	❗A6 Demo (2/4)	-	Ethics and Accessibility
10	Wednesday, March 15 2023	-	Work in Class	-
11	Wednesday, March 22 2023	-	Work in Class	-
12	Wednesday, March 29 2023	❗A6 Demo (3/4)	Last day of class - Demos of your Software	-
-	Friday, March 31 2023	❗Bonus Due	-	-
13	Wednesday, April 5 2023	❗A6 Demo (4/4)	Work in Class, Final Lecture TBD	-
-	Wednesday, April 12 2023	❗A7 Due ❗Participation Comment Due	Software Due, Participation Comment Due	-

Notes

- *Lectures subject to change.*
- Zoom link will be provided in the LearnSoftware App. You will be given access via email close to the start of the course.
- Not all lectures listed here have a corresponding entry in the presentations page on this site. Some exist as videos only on the LearnSoftware App.

Summary

- Lectures are asynchronous via the Learn Software application
- We will try to bring in some guest speakers
- 1 AI + Data Tutorial
- 2 discussions (1 includes short demos)
- 4-5 class working periods
- 1 Intro
- 4 Demo periods (part class each time)

Your software

To help you keep on track, here are some general recommendations and guidelines on milestones and goals you should aim for throughout the term. While most of these aren't mandatory (unless enforced by assignments), following these will help ensure your success and avoid stress and chaos in the last 2 weeks of the course.

- You should start a skeleton hello world application as soon as you choose your base tech (Flask, Rails, iOS, etc)
- You should start building out a developer environment, CI setup, test infra, and production setup by the end of January
- You should start building out functionality no later than mid-February
- Your models should also be started no later than mid-February
- You should have demoable content by the first week of MArch
- You should have significant demoable content by the last demo
- You should be at the point of adding no new major features in April. This should be reserved for polish and bug fixes

Demos

Here is an example of the progress I would expect for each demo slot:

- Demo 1: Decisions and Tech Stack, at minimum a Hello World application in your chosen language/platform
- Demo 2: Your application running on CI, which means some basic tests (at minimum). It should also be launched onto the web / mobile
- Demo 3: Minimal front end, mobile, and back end (as applicable). Starting code for any data models. Some functionality
- Demo 4: Full software, general functionality, note any shortcomings or areas to improve. Identify all main use cases implemented

Assignments

Please see the assignments section.

Class Policies

These are the policies for this course.

- Attendance
- Communications
- Late Policy
- Midterms & Exams (hint: there are none)
- Plagiarism
- Audio + Video Recording
- Support & Accommodation
- Team Grades
- Repos

Support and Accommodations

Students with diverse learning styles and needs are welcome in this course. Students who have a disability or health consideration that may require accommodations are both encouraged and welcome to approach the Course Instructors as soon as possible. Should accommodations be necessary, by University of Toronto policy students are required to contact the Accessibility Services Office.

Communication

Expectations

You are senior/graduate-level Computer Science students. As such, you are expected to have a high standard of communication. You need to be able to communicate clearly and effectively in a team setting, but as importantly - concisely and effectively communicate your needs with the teaching staff.

You will see a theme throughout this course centred around communication. This is a very important skill that you will need to develop in order to be successful in your career.

Some Explicit Expectations

- Your team will have a Standup configured on LearnSoftware App and team members will be expected to respond to it at least once per week.
- You will use GitHub Issues to communicate with your team and show your work throughout the term
- Missing demos and other aspects of the class can and will be penalized *if you do not communicate in advance*. I cannot stress this enough - this class is very accommodating of your time and schedule, granted you respect the time of others (including the teaching staff).
- The class is not that large, I know who is attending lectures. I expect you to be able to communicate your needs - are you going to be late due to work? Are you going to be late because you have a class on the other side of campus? Talk to me, and we it won't be an issue.

Methods of Communication

There are, just as you will find in a startup, various methods of communication. Please hold group online discussions on Quercus, Slack or another tool of your choosing. You will receive invites during the first week of the course. Likewise, Zoom will be used at points during this course. Please ensure you have it installed and signed into an account.

Communication, as we will discuss in the course, is vital to the success of any business. As your startup gains traction and becomes successful, you will find that communication methods and culture become a bottleneck within your company. As such, we will ensure students are exposed to different means of communication and include the following table to help you understand the various mediums.

Medium	When to use
LearnSoftware App	Creating teams. Lessons. Creating Repos. Managing Repos. Assignment feedback, surveys, etc for CSC491/2600
Slack	General discussions and questions. The instructor for CSC491/2600 (Julian Nadeau) also prefers this communication method.
Email	Private matters. The instructor for CSC454/2527 (Mario Grech) also prefers this communication method.
Quercus	CSC454/4527 uses Quercus
In Person	Hallway conversations. If you decide anything here, write it down somewhere else on your Team's GitHub Repo
Video Chat via Zoom	All classes will happen over Zoom
Issues and tracking boards	Team: Use this heavily as a decision record on most topics. Making use of GitHub Issues shows participation from all users (including non-technical) and will be a part of your grade in

Medium	When to use
	CSC454/2527 CSC491/2600: When you have a problem that needs to be fixed and are comfortable talking about it in the open
GitHub Releases	For your team repository, use this as a method to record assignment submission for CSC491/2600
Internal Wikis	You can keep team docs in your team repo on GitHub. I ask that you avoid using the wiki feature as it makes grading more difficult
GitHub Issues	You are required to use this for roadmapping and planning

Midterms & Exams

There are none. This is a project-based course.

Class Attendance

While class attendance is mandatory, we will not be taking attendance. This is a senior computer science class with a highly competitive application process. As such, you are trusted to make the best choice for yourself and your team.

However, as stated in the team grade policy, team members all receive the same grade unless there is an obvious discrepancy in the output of work, then as per policy we will be forced to give you a different grade.

Attendance may be taken into account, specifically if you are consistently late/absent and you cannot work with your team (without talking to the instructor prior to class) you may see your team mark affected.

You must also remember that you are in a team. Your team members are counting on you to do your part. Don't let them down.

I am sick or injured, have an emergency, or family issue

Life happens. I am not going to penalize you for things that are outside of your control. Instead, I ask that you be open and communicate your needs with me whether that is lenience on an assignment deadline or that you may not attend class.

If you are sick and would like to participate in class still, then we can set up a Zoom call for your to join in remotely. *Please do not* come to class if you are sick. I *will not* penalize you for not attending class due to an illness. Be open and communicate your needs so we can accommodate.

All my classes are online and I do not live in a timezone that can align with EST

Please talk to the instructor by email or Slack.

How to ask for help

- A ping on Slack is a decent option
- An email leaves a better paper trail for both of us
- In some cases, meeting in person may be a good option (we'll discuss this beforehand)

While I do not need to know the exact details of what happened, just let me know what the issue is and how long you need to recover/return to normal work.

However, for extended absences or issues that persist past the end of the semester we may need to ask the University's administration for input. We can work through that one together.

Video + Audio Recording

This course, including your participation, will be recorded on video and will be available to students in the course for viewing remotely and after each session.

Course videos and materials belong to your instructor, the University, and/or other source depending on the specific facts of each situation, and are protected by copyright. In this course, you are permitted to download session videos and materials for your own academic use, but you should not copy, share, or use them for any other purpose without the explicit permission of the instructor.

For questions about recording and use of videos in which you appear please contact your instructor.

Student Recording

Students may create audio-recordings of the lectures for their personal use. Recordings are intended to permit lecture content review so as to enhance understanding of the topics presented. Audio-recordings are not substitutes for attending class.

Students should note that since audio recordings are to be permitted, their voice may be recorded by others during the class. Please speak to the instructor if this is a concern for you.

In accordance with the Accessibility for Ontarians with Disabilities Act, 2005, persons who have special needs will be accommodated.

Students agree to the following terms when creating audio recordings of lectures:

- Recordings are not to be distributed without the permission of the instructor via the Internet, using social media such as Facebook, peer-to-peer file sharing such as One Drive or Dropbox, or other distribution channels.
- Recordings are not to be shared with other classmates unless they are to be used in collaborative assignments, or if the instructor permits for other reasons.

Non-compliance with these terms violates an instructor’s intellectual property rights and the Canadian Copyright Act. Students violating this agreement will be subject to disciplinary actions under the Code of Student Conduct.

Team Grades

When working in teams, students are expected to divide workload equitably. The nature of the division is up to the team members, and does not require that all members work the same hours or produce identical volumes of work. By default all team members receive an identical grade on team assignments, except for any part noted as "individual".

During your first assignment you will create a "team expectations" document that all team members must agree on. This will help us set expectations for workload and expectations of all members.

Students should report any difficulties in their teams to a member of the Teaching Team as early as possible so that the difficulties can be addressed in a positive way.

Students should also maintain as complete a record of team interactions as possible. GitHub Issues are the best way to accommodate this, and are required to be used for roadmapping and gathering team feedback.

Based on solicited, confidential feedback, or at the instructors discretion based on participation in the course - the Course Instructor may adjust the grade distribution within a team through means including lower participation grades and overall adjustment of the grade for individual assignments.

Plagiarism

The University of Toronto treats plagiarism as a violation of the Code of Behaviour on Academic Matters. Plagiarism is a serious form of cheating in which a student makes use of someone else's ideas or words without giving appropriate attribution. In your academic work, plagiarism usually occurs in one of three ways:

- You cut and paste a piece of someone else's text or code or figure but do not clearly show what the source is for that material.
- You hand in work done by others (e.g. teammates) without putting their names on the work.
- You re-phrase someone else's idea into your own words, but do not give credit to the source of the idea.

The University takes cheating very seriously. Penalties can include zero on the assignment, zero in the course, annotations on your transcript (which would be seen by a potential graduate school or employer), or in extreme cases expulsion from the University. If you are concerned about your use of sources, discuss your concerns with your Course Instructor before submitting a document for assessment.

Late Policy

Students are expected to work diligently to pass their assignments in on time. This course is intended to partially model a startup, however it is still a university course. Assignments also take time to mark and lecturers/TAs schedule their time according to the course calendar. We ask that you be respectful of their time by not passing assignments in late.

Mark Deductions

Assignments will be accepted up to 5 days past the due date at -10% per day.

Days Late	Percent Lost
1	-10%
2	-20%
3	-30%
4	-40%
5	-50%
6+	-100%

NOTE: This policy is slightly different for synchronous demos. Please see that assignment for further details on the late policy.

Asking for an Extension

- Accommodations can be made by talking to the instructor. They are not guaranteed, however, but we do like to model a startup so they usually are acceptable! :)
- Extensions that are requested within 6 hours of the due date will incur a penalty of 1 day, so please do not wait until the last minute to communicate with the instructor.

Compounding assignments

These assignments are, generally, made to compound one another. While assignments may be late and you may lose 100% of the marks, you must still complete them to work on the following assignments.

Repositories

- We use https://github.com/dcsil
- You will have a repository automatically made for team notes, profiles, and other team metadata. This repo **is not for code**. You will lose 20% on each assignment where you submit code via this repo.
- You can create as many repos as you need by going to https://learnsoftware.engineering/my_team/new_repo
- You can see all your team's repos by going to https://learnsoftware.engineering/my_team/repos
- Every repository must have a `service.yml` that *passes* checks. Your service.yml is validated on push, however it is also shown on the LearnSoftware app.
    - Every repository must therefore have logging, exception handling (via Sentry), a bootstrap script etc