Welcome!

CS 346 Application Development

https://student.cs.uwaterloo.ca/~cs346

Introduction

Instructors

- Dr. Jeff Avery
- Dr. Adrian Reetz

Instructional Support Coordinator

- Caroline Kierstead

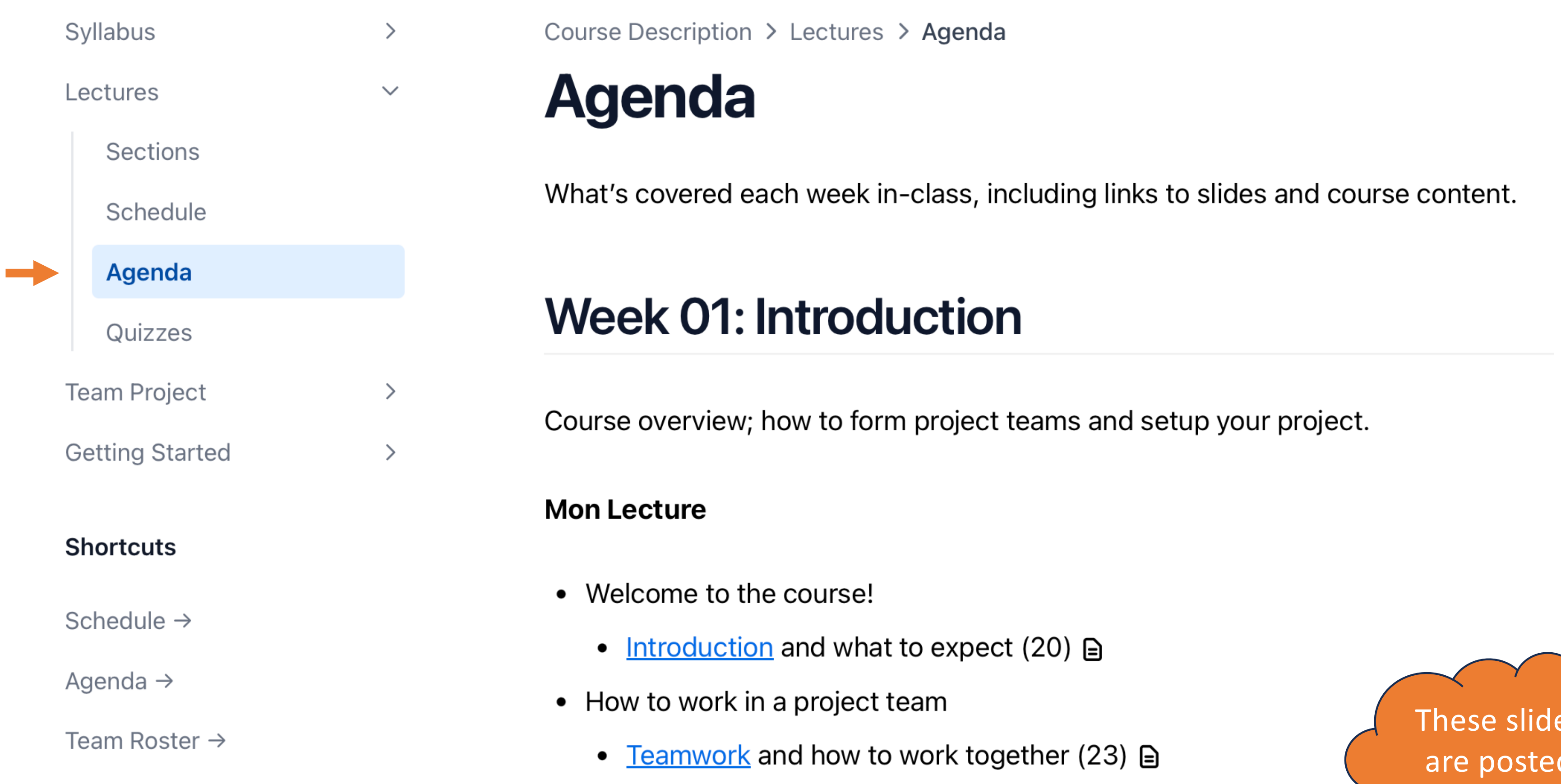
Teaching Assistants

- Samiha, Paul, Christopher, Amber, Eric, + 1

See course website for contact information

Prof. Avery

Prof. Reetz



Agenda

What's covered each week in-class, including links to slides and course content.

Week 01: Introduction

Course overview; how to form project teams and setup your project.

Mon Lecture

- Welcome to the course!
  - Introduction and what to expect (20) 🗎
- How to work in a project team
  - Teamwork and how to work together (23) 🗎

These slides are posted here

https://student.cs.uwaterloo.ca/~cs346

Course Overview

What are we going to be doing this term?

Course Description

CS 346 Application Development
LAB, LEC, TST 0.50

Introduction to **full-stack application design and development**. Students will work in **project teams** to design and build complete, working applications and services using standard tools. Topics include **best practices** in design, development, testing, and deployment.

Prerequisites: CS 246; Computer Science students only.

https://student.cs.uwaterloo.ca/~cs346/1261

What is this course about?

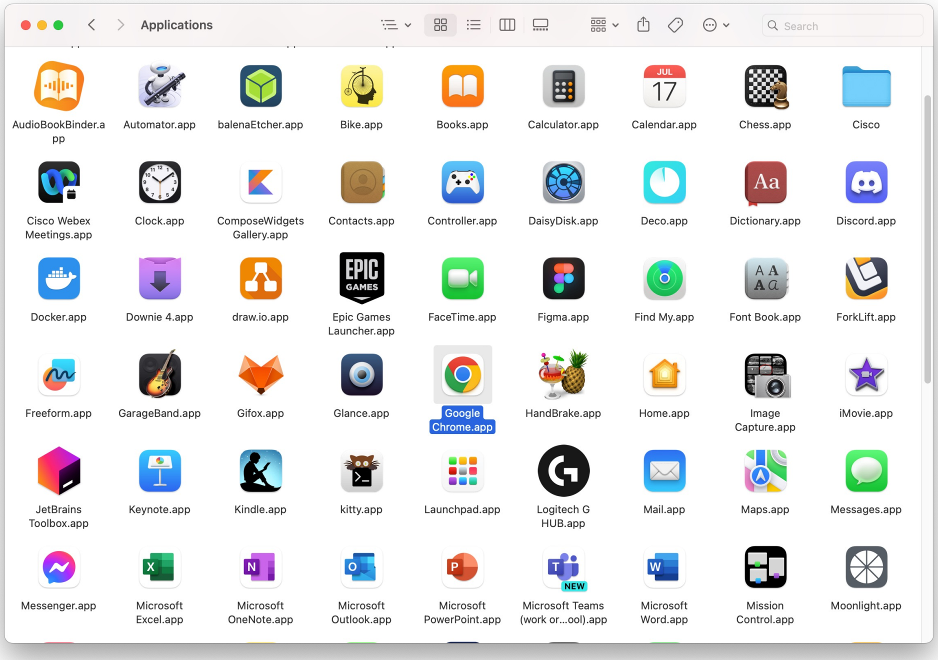
This is a course about building “applications” i.e., software that solves problems for people. Most of the software that you use on day-to-day basis is **application software**.

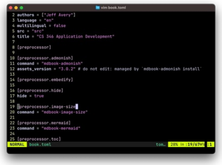
Application software tends to be:

- Focused on providing value for a person.
- Interactive and graphical.
  - e.g., calculator, MS Word, Fortnite.
- Tends to be very personalized.
  - Installed on personal devices.
  - Save profiles/preferences.
  - Personal data (confidential?)

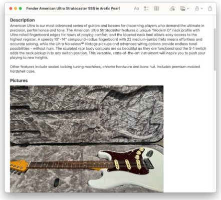
Contrasts with other types of software:

- Embedded -- controls hardware systems.
- System software -- provides services for other software.





```
authors = ["Jeff Avery"]
language = "en"
multilingual = false
src = "src"
title = "CS 346 Application Development"

[preprocessor]

[preprocessor.admonish]
command = "mdbook-admonish"
assets_version = "3.0.2" # do not edit: managed by `mdbook-admonish install`

[preprocessor.embedify]

[preprocessor.hide]
hide = true

[preprocessor.image-size]
command = "mdbook-image-size"

[preprocessor.mermaid]
command = "mdbook-mermaid"

[preprocessor.toc]
NORMAL  book.toml                                    tom…  28% ln :19/67≡% :1
```

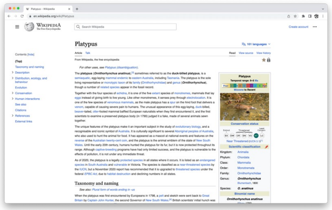


Description

American Ultra is our most advanced series of guitars and basses for discerning players who demand the ultimate in precision, performance and tone. The American Ultra Stratocaster features a unique "Modern D" neck profile with Ultra rolled fingerboard edges for hours of playing comfort, and the tapered neck heel allows easy access to the highest register. A speedy 10"-14" compound-radius fingerboard with 22 medium-jumbo frets means effortless and accurate soloing, while the Ultra Noiseless™ Vintage pickups and advanced wiring options provide endless tonal possibilities - without hum. The sculpted rear body contours are as beautiful as they are functional and the S-1 switch adds the neck pickup in to any switch position. This versatile, state-of-the-art instrument will inspire you to push your playing to new heights.

Other features include sealed locking tuning machines, chrome hardware and bone nut. Includes premium molded hardshell case.

Pictures



Platypus

From Wikipedia, the free encyclopedia

For other uses, see Platypus (disambiguation).

The **platypus** (***Ornithorhynchus anatinus***),[3] sometimes referred to as the **duck-billed platypus**, is a semiaquatic, egg-laying mammal endemic to eastern Australia, including Tasmania. The platypus is the sole living representative or monotypic taxon of its family (Ornithorhynchidae) and genus (*Ornithorhynchus*), though a number of related species appear in the fossil record.

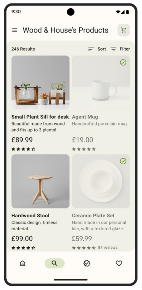
Together with the four species of echidna, it is one of the five extant species of monotremes, mammals that lay eggs instead of giving birth to live young. Like other monotremes, it senses prey through electrolocation. It is one of the few species of venomous mammals, as the male platypus has a spur on the hind foot that delivers a venom, capable of causing severe pain to humans. The unusual appearance of this egg-laying, duck-billed, beaver-tailed, otter-footed mammal baffled European naturalists when they first encountered it, and the first scientists to examine a preserved platypus body (in 1799) judged it a fake, made of several animals sewn together.

The unique features of the platypus make it an important subject in the study of evolutionary biology, and a recognisable and iconic symbol of Australia. It is culturally significant to several Aboriginal peoples of Australia, who also used to hunt the animal for food. It has appeared as a mascot at national events and features on the reverse of the Australian twenty-cent coin, and the platypus is the animal emblem of the state of New South Wales. Until the early 20th century, humans hunted the platypus for its fur, but it is now protected throughout its range. Although captive-breeding programs have had only limited success, and the platypus is vulnerable to the effects of pollution, it is not under any immediate threat.

As of 2020, the platypus is a legally protected species in all states where it occurs. It is listed as an endangered species in South Australia and vulnerable in Victoria. The species is classified as a near-threatened species by the IUCN, but a November 2020 report has recommended that it is upgraded to threatened species under the federal *EPBC Act*, due to habitat destruction and declining numbers in all states.

Taxonomy and naming

Platypus	
Temporal range: 9–0 Ma	
Miocene to Recent	
Conservation status	
Near Threatened (IUCN 3.1)[1]	
Scientific classification	
Kingdom:	Animalia
Phylum:	Chordata
Class:	Mammalia
Order:	Monotremata
Family:	Ornithorhynchidae
Genus:	*Ornithorhynchus*
	Blumenbach, 1800



Small Plant Sill for desk Beautiful made from wood and fits up to 3 plants! £89.99	Agent Mug Handcrafted porcelain mug £19.00
Hardwood Stool Classic design, timless material. £99.00	Ceramic Plate Set Hand made in our personal kiln, with a textured glaze. £59.99 84 reviews

1950s - 1960s

Console

C++ (Python, C)

Imperative

1970s – 1980s

Desktop

Pascal, C++, VB

Object-Oriented

Graphics

Databases

1990s– 2000s

Web

Java, JS, HTML/CSS

Functional

Databases

Networking

Concurrency

2010s – 2020s

Mobile

Kotlin, Swift

OO & Functional

Graphics

Databases

Networking

Concurrency

We will support 2 or 3 of these styles; web applications are covered in CS 349.

What are you specifically doing?

We will discuss requirements in more detail in Week 2

You will form teams and pick your own project.

- Team of 4 people
- You will pitch a proposal to us, design and implement it.

It must be a mobile or desktop application – *no webapps*.

- Needs to be graphical and be *moderately complex*.
- Includes local and remote functionality e.g., online DB.
- Includes features that you define to *solve a particular problem*.

You will be given technical guidelines i.e., what language/toolkits.

- Some flexibility to choose suitable libraries for your application.
- You will need to learn a new language/toolchain! i.e., Kotlin.

Do we *have* to work on a team?

Yes! There's a lot of reasons for this.

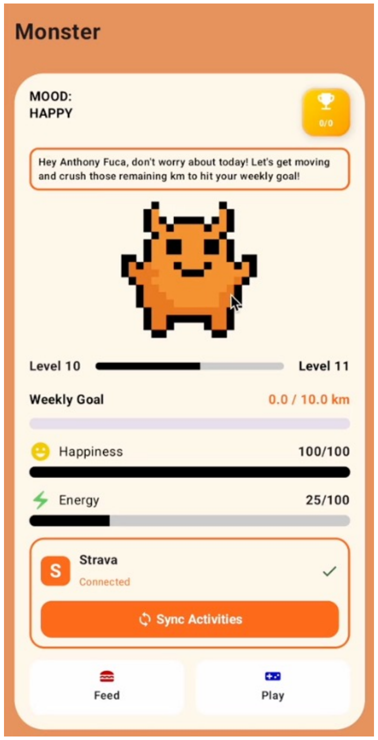
- You can build something a lot more interesting with a team of people. There is simply too much to do for a single person to pull it off.
- It's much less risky to have 4 people working on a project e.g., get sick? something still gets delivered.
- This provides a chance for you to gain technical depth since you can divide up the work across the team e.g., BE vs FE development.
- The development practices that you use are different when you are part of a team. e.g., branching, merging, writing meaningful documentation. *These are core software development skills that only come about on a team*.
- It's a chance to improve teamwork and leadership skills. *Employers care about team skills*.

Past Projects

Every term, teams produce **outstanding** projects.

Examples from last term

- Runster: *collects data and gamifies running.*
- SmartReceipt*: mobile app to capture, organize, and analyze receipts.*



Monster

MOOD:
HAPPY

0/0

Hey Anthony Fuca, don't worry about today! Let's get moving and crush those remaining km to hit your weekly goal!

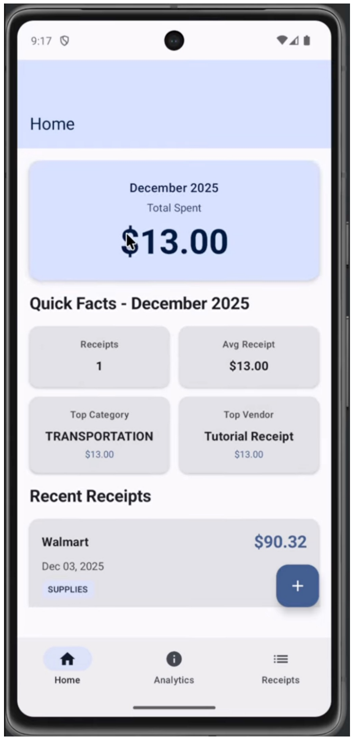
Level 10 Level 11

Weekly Goal 0.0 / 10.0 km

Happiness 100/100

Energy 25/100

Strava
Connected ✓



9:17

Home

December 2025
Total Spent
$13.00

Quick Facts - December 2025

Receipts 1	Avg Receipt $13.00
Top Category TRANSPORTATION $13.00	Top Vendor Tutorial Receipt $13.00

Recent Receipts

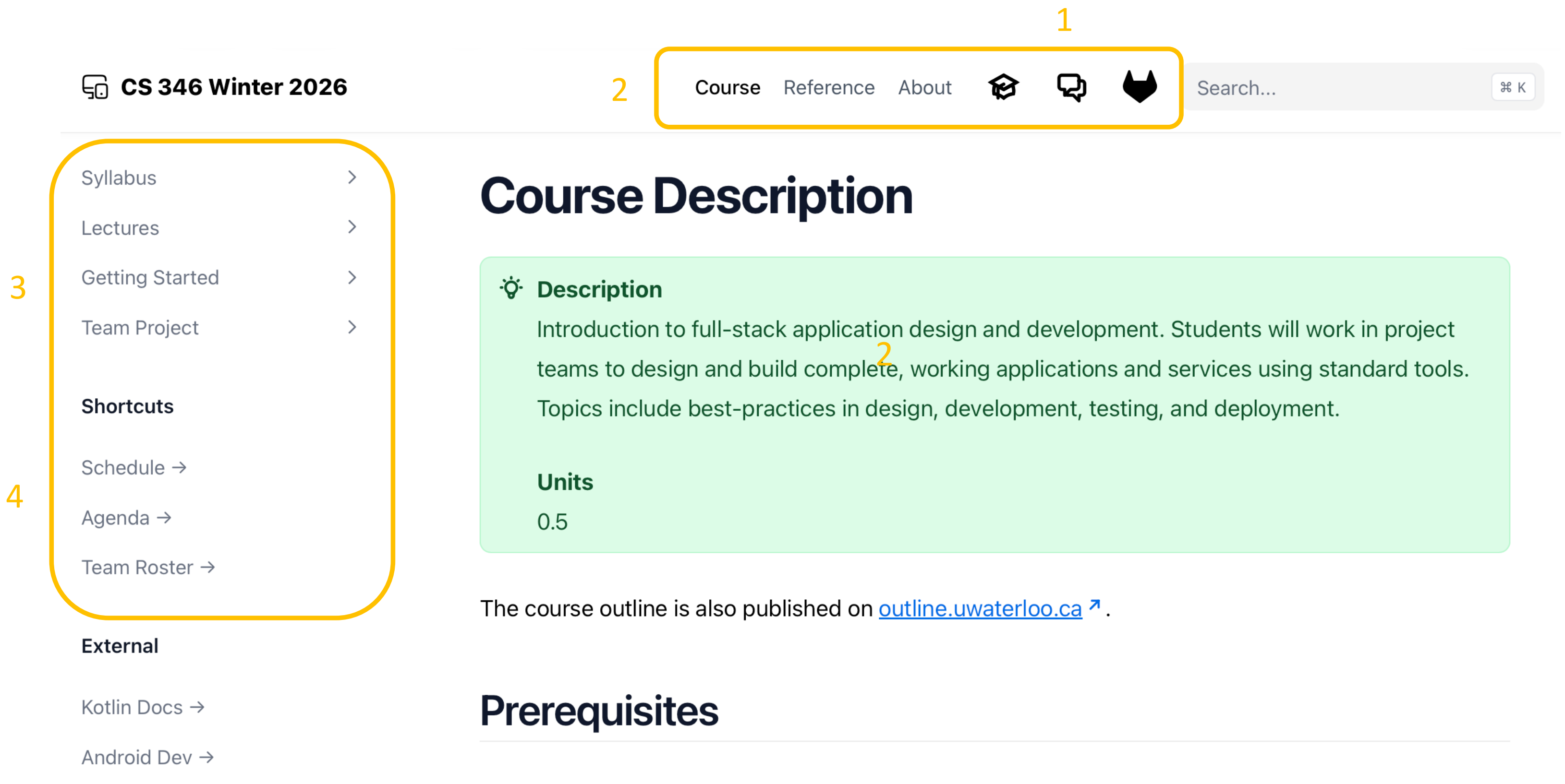
Walmart $90.32
Dec 03, 2025
SUPPLIES

https://student.cs.uwaterloo.ca/~cs346/1261/course/project/project-gallery/

Navigating the course

Let’s discuss how the course is put together.

Demo the course website



Course Description

Description

Introduction to full-stack application design and development. Students will work in project teams to design and build complete, working applications and services using standard tools. Topics include best-practices in design, development, testing, and deployment.

Units

0.5

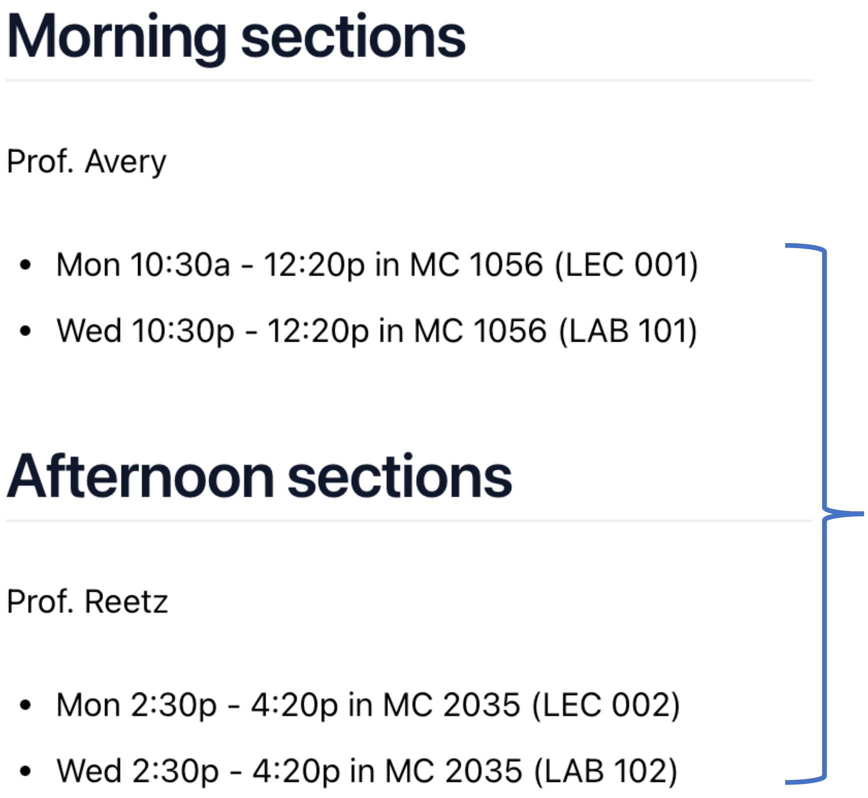
The course outline is also published on outline.uwaterloo.ca ↗ .

Prerequisites

https://student.cs.uwaterloo.ca/~cs346/1261/course/

Class Structure

This is illustrative.
You can *generally*
expect more free
time on Wed



Morning sections

Prof. Avery

- Mon 10:30a - 12:20p in MC 1056 (LEC 001)
- Wed 10:30p - 12:20p in MC 1056 (LAB 101)

Afternoon sections

Prof. Reetz

- Mon 2:30p - 4:20p in MC 2035 (LEC 002)
- Wed 2:30p - 4:20p in MC 2035 (LAB 102)

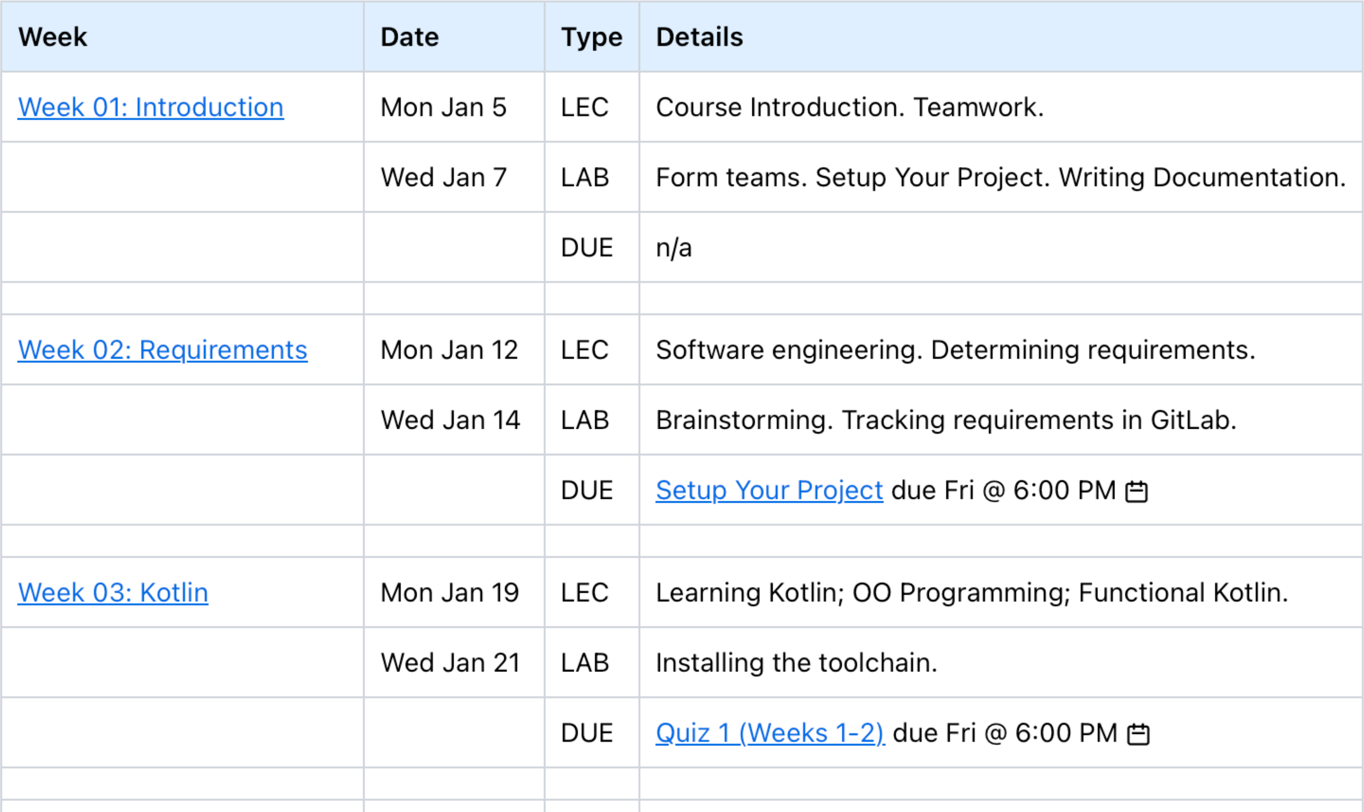
Mon LEC section

- 1h 20m lectures + 30 min free
- Regular lectures & topics.
- Nothing due.

Wed LAB section

- 30m lectures + 1h 20m free
- Work on your project w/ your team.
- Instructor + TAs present
- Demos during **some** LABs

Lectures



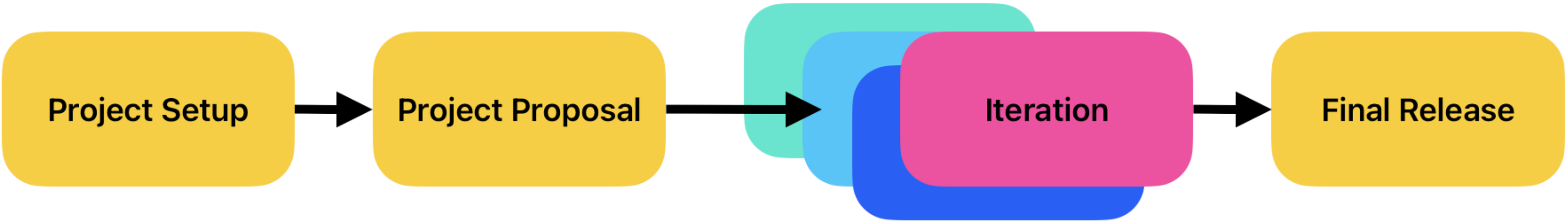
Week	Date	Type	Details
Week 01: Introduction	Mon Jan 5	LEC	Course Introduction. Teamwork.
	Wed Jan 7	LAB	Form teams. Setup Your Project. Writing Documentation.
		DUE	n/a
Week 02: Requirements	Mon Jan 12	LEC	Software engineering. Determining requirements.
	Wed Jan 14	LAB	Brainstorming. Tracking requirements in GitLab.
		DUE	Setup Your Project due Fri @ 6:00 PM 📅
Week 03: Kotlin	Mon Jan 19	LEC	Learning Kotlin; OO Programming; Functional Kotlin.
	Wed Jan 21	LAB	Installing the toolchain.
		DUE	Quiz 1 (Weeks 1-2) due Fri @ 6:00 PM 📅

1h 30m + open time

20m + open time

See schedule for the complete list of topics.

Project Structure



Deliverable	What is it?	Date
Project Setup	Form your team, setup your project space.	Week 02
Project Proposal	Pitch your product idea to your TA!	Week 04
Sprint 1	User interfaces implemented (semi-functional)	Week 06
Sprint 2	Domain layer implemented, UI functional (mocked)	Week 09
Sprint 3	Data layer implemented, some features complete.	Week 11
Sprint 4	Final release, all features complete.	Week 13
Final release	Final software package + documentation submitted.	Week 14

Assessment

Most of your grade comes from team activities.

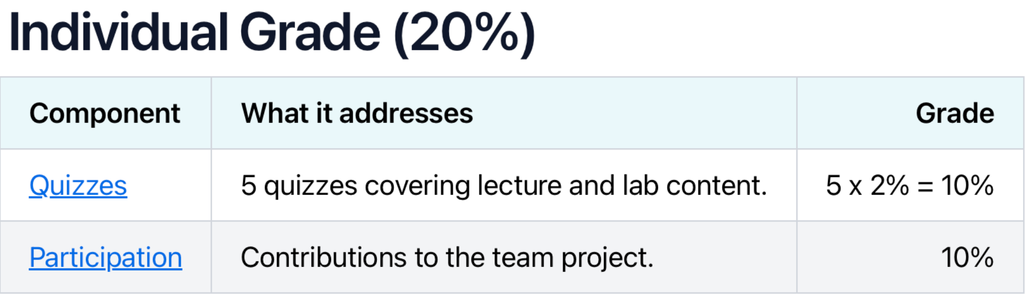
Deadlines every 2 weeks

- Demo to your TA.
- Receive feedback.
- Submit your code/tests.

Everyone participates!

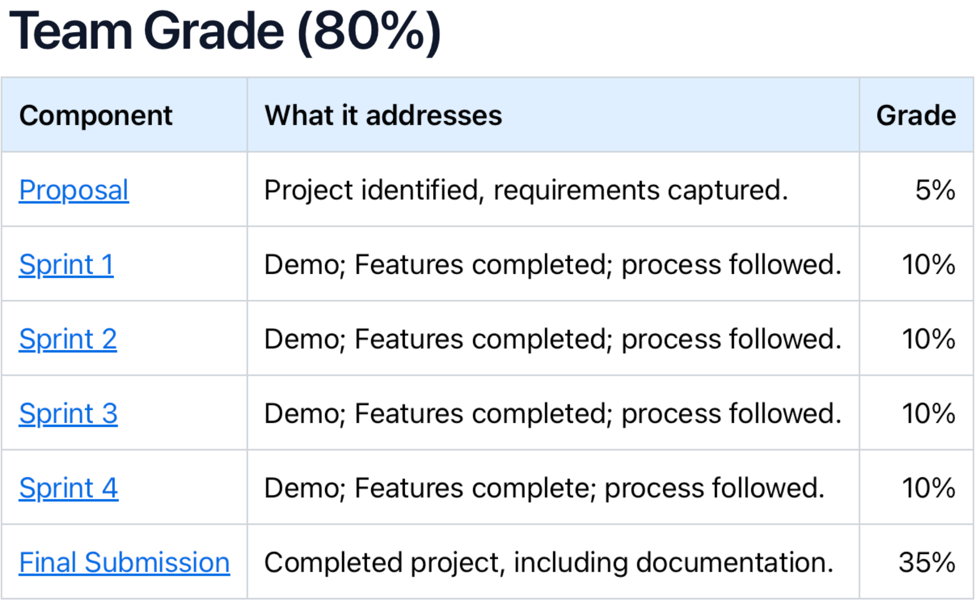
- Zero for missing a demo.
- See Illness vs STA policies.

See assessment details.



Individual Grade (20%)

Component	What it addresses	Grade
Quizzes	5 quizzes covering lecture and lab content.	5 x 2% = 10%
Participation	Contributions to the team project.	10%



Team Grade (80%)

Component	What it addresses	Grade
Proposal	Project identified, requirements captured.	5%
Sprint 1	Demo; Features completed; process followed.	10%
Sprint 2	Demo; Features completed; process followed.	10%
Sprint 3	Demo; Features completed; process followed.	10%
Sprint 4	Demo; Features complete; process followed.	10%
Final Submission	Completed project, including documentation.	35%

General Policies

- **Illness**
  - You must submit a VIF, email the instructor, and coordinate with your team.
  - A missing component grade will be redistributed across other components.
- **Short-term absences**
  - STA cannot be used for team deliverables i.e., demos, final submission. It can be used for quizzes, which would shift the weight of the quiz you missed.
- **Exams & INC**
  - This course has no final exam and an INC will never be granted for missing team deliverables. Other policies would apply – *see next slide*.

https://student.cs.uwaterloo.ca/~cs346/1259/course/syllabus/policies/

Team Policies

- **You must form teams by the end of week 2.**
  - We will help, but you are responsible for finding a team.
  - You will have time after-class to match with teams. See Piazza as well.
- **You cannot take this course while on a (remote) work term.**
  - You must be physically present to meet with your team & attend class.
  - You *cannot* take it remotely and "call in" for demos.
- **You need to physically attend and participate in team demos.**
  - You will receive a grade of zero for a deliverable if you don't participate in the demo.
  - In extreme cases, we can remove you from the course or adjust your final grade (down) if you repeatedly fail to engage.

Code Policies

- **“Borrowed Code”**
  - You may use external code (up to 25 lines) with appropriate citation
  - You cannot use projects from previous offerings of this course.
  - You cannot use any portion of a project from a different course
    - e.g., CS 446 or CS 449 are also project courses. You cannot normally submit the same project in two different courses.

- **Generative AI / LLMs**
  - Can be used for code analysis. e.g., ”Gemini, what does this code do?”
  - If used to generate code, it should be treated like any other source. i.e., must be documented, subject to restrictions like any other citation.
  - Cannot use an LLM to generate more than this!

Getting Help?

- Come to class and work on your project!
  - Mon: the instructor is there
  - Wed: the instructor AND YOUR TA is there
- Piazza
  - Link at top of the website.
  - I’ll reply within a few hours (M-F, 9-5). Weekends may be longer.
- Reference section of the website
  - Expanded documentation, tips.