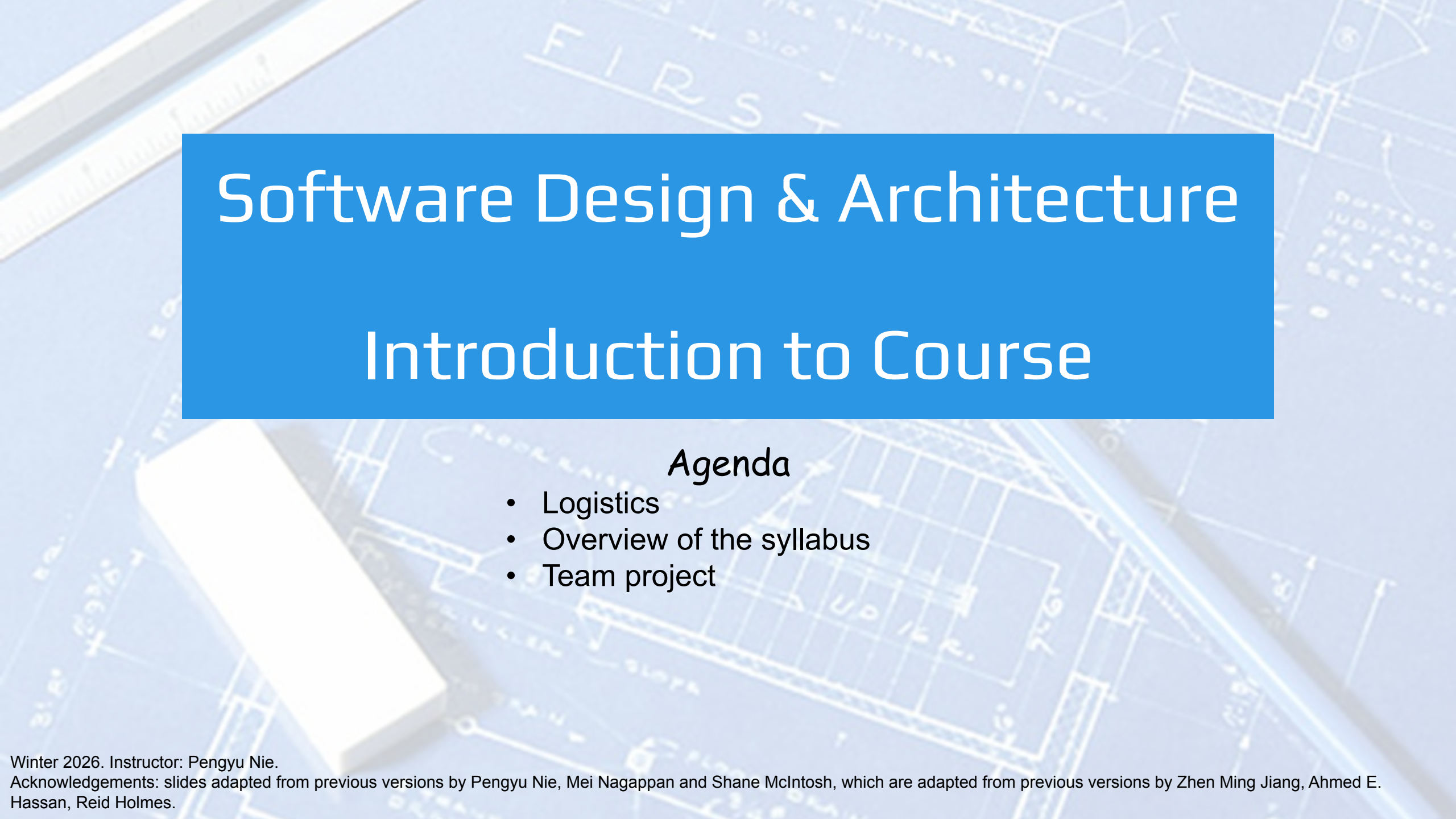




Software Design & Architecture

Introduction to Course

Agenda

- Logistics
- Overview of the syllabus
- Team project

Who am I?



Lesley Istead

Adjunct Assistant Professor (UW CS)

- UW BMath CS, MMath CS, PhD (2001 - 2018)
- ***not actually Zelda***
- lesleyistead@gmail.com ← please use **THIS** email

Research Areas

- AI (benchmarking, applications, UI/UX)
- computer graphics, VR, S3D, games

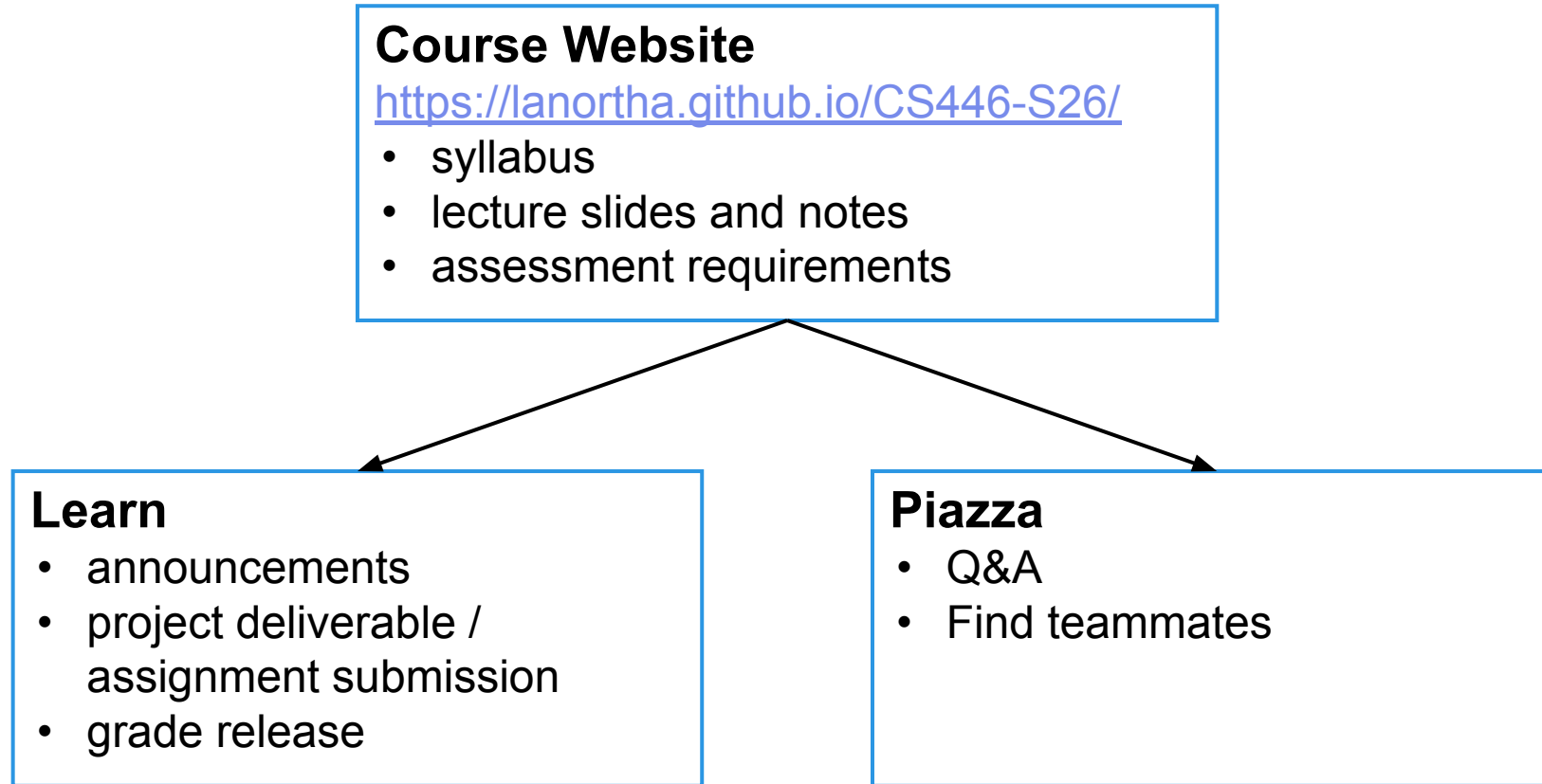
Industry

- Software Engineer: Google, Epson, Sun Microsystems
- Research: SideFX Software, Deluxe, Gener8, Adobe

Your TAs

- Taha Cheema <t2cheema>
- Albert Shi <g7shi>
- Vincent Wang <w278wang>
- How will they assist you?
 - Each project team will be assigned to one TA, who will provide technical assistants and feedback throughout the term
 - Meet them during workshop sessions; additional office hours by appointment

Key Information Source



Dates and Times

- Sec 001: MW 1PM-2:20PM @ MC 2054
- Sec 002: MW 2:30PM-3:50PM @ MC 20
- Check the **syllabus** for project deadlines
 - All deadlines refer to 11:59pm Eastern Time on that day
 - Presentations are in class
- **Announcements** via Learn (archived) and emails (as long as you didn't filter emails from Learn)
 - Reminders of deadlines
 - Any change in lecture topics / project deadlines

Communication Rules

- **Piazza**

- Technical questions, logistic questions, generic project-related questions
- Use folders to categorize the question
- Feel free to try answering others' questions
- First two weeks: post to find teammates

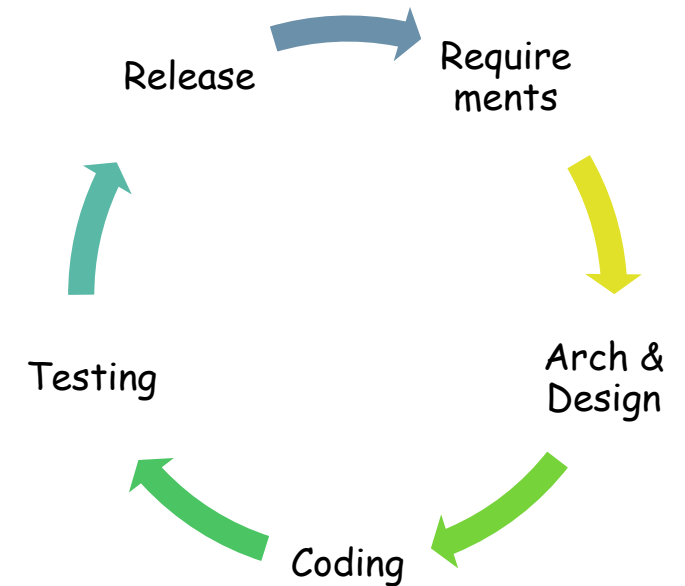
- **Email**

- More private questions (e.g., personal or specific to your project team)
 - Prefix your email title with [CS446] or [ECE452] or [CS646]
 - CC all team members (stakeholders) if project related
- Please do NOT leave your questions to the last minute. We may not respond to requests made within 24h before the deadline.

Syllabus

Workshops

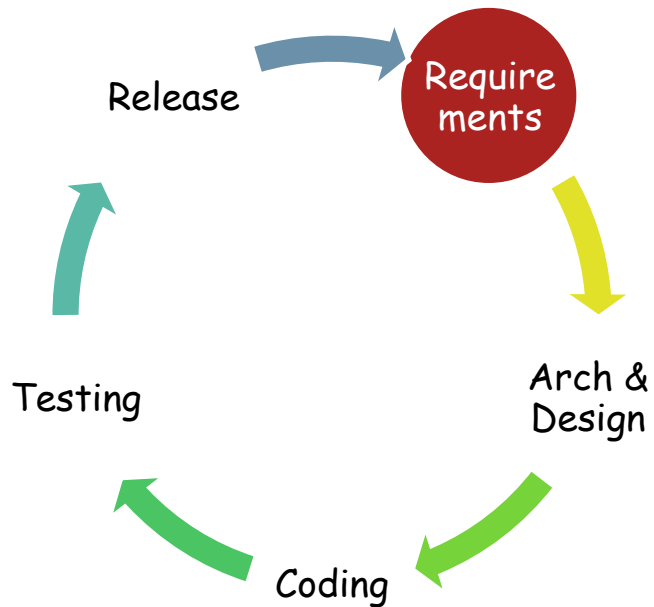
- Interviews and Stakeholders
- Android Toolchain
- Jetpack Compose
- Testing
- Continuous Integration
- Release Engineering



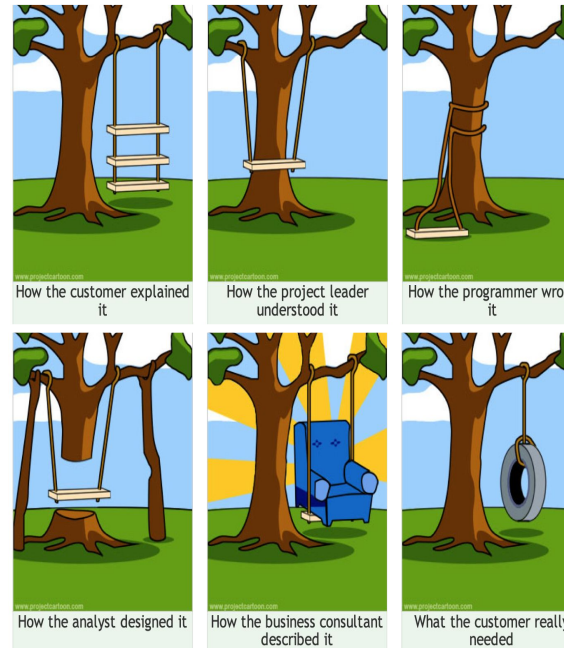
Course Delivery

- Lecture
 - theoretical concepts about software design and architecture
 - mostly your instructor presenting, mixed with some discussions
- Workshop
 - (~20min) tutorials or demonstrations
 - (~1h) team work on your project
 - course staff will be available for consultations

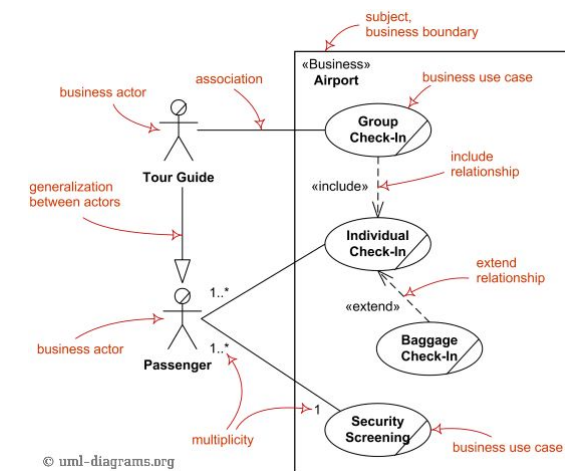
Overview of Syllabus: Requirement Phase



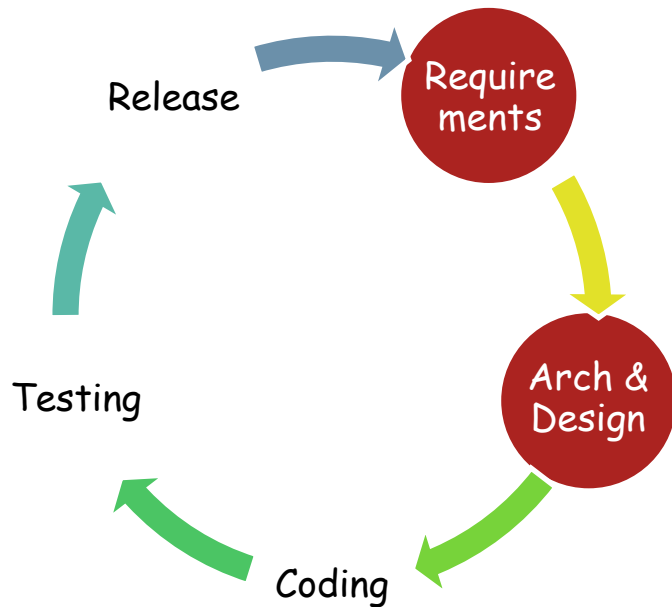
- Intro to Course
- Intro to Software Architecture, Stakeholders
- Non-Functional Requirements
- User Scenarios, Use Cases, Human Values
- Software Modelling, UML Diagrams
- Decomposition, Architectural Views
- Architectural Styles 1 and 2
- Design Patterns 1 and 2
- Evolution of Software Design and Architecture



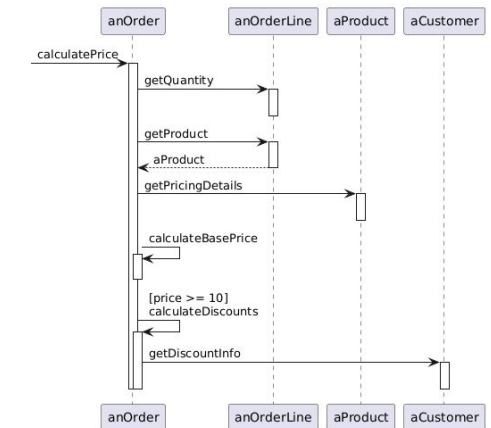
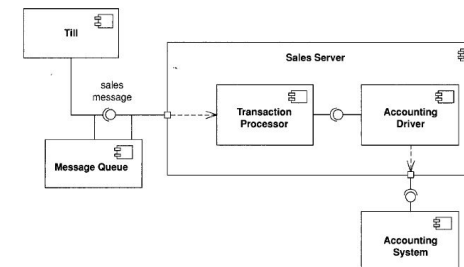
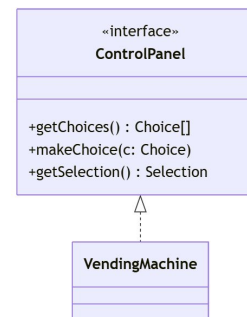
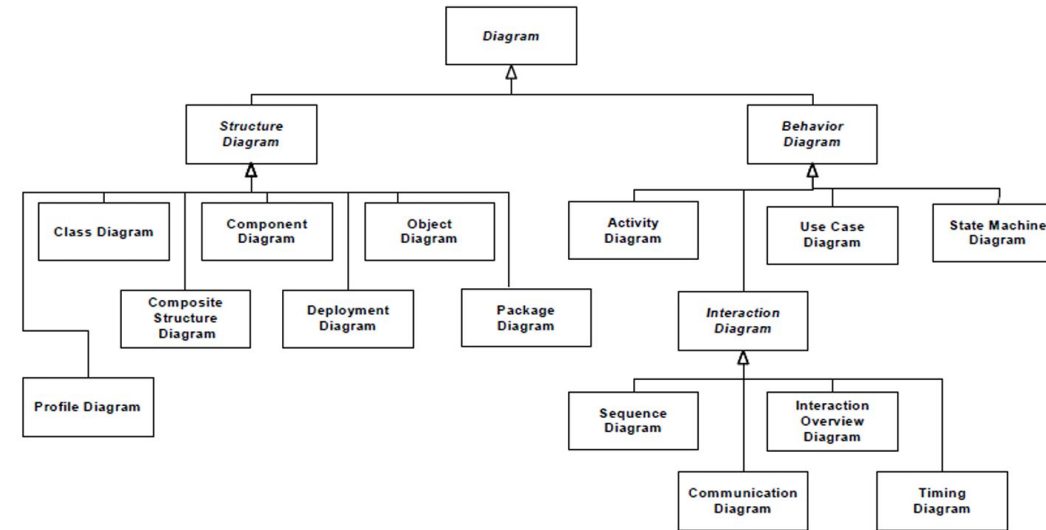
Efficiency	Privacy	Availability	Complexity
Usability	Security	Reliability	Readability
Portability	Survivability	Robustness	Heterogeneity
Accessibility	Safety	Fault-tolerance	
		Scalability	
		Evovability	



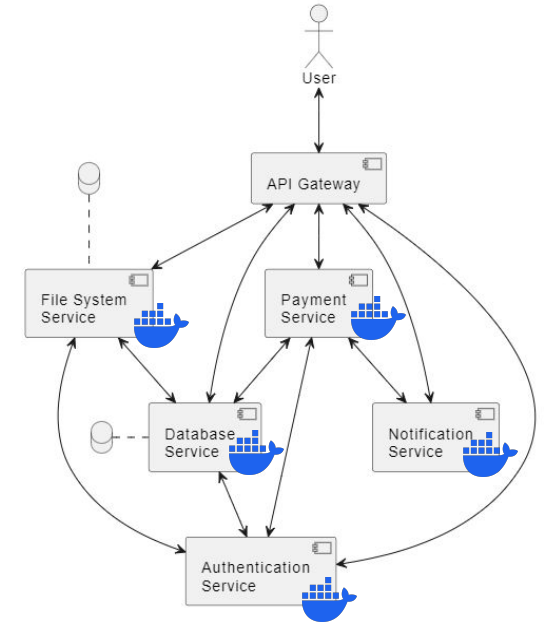
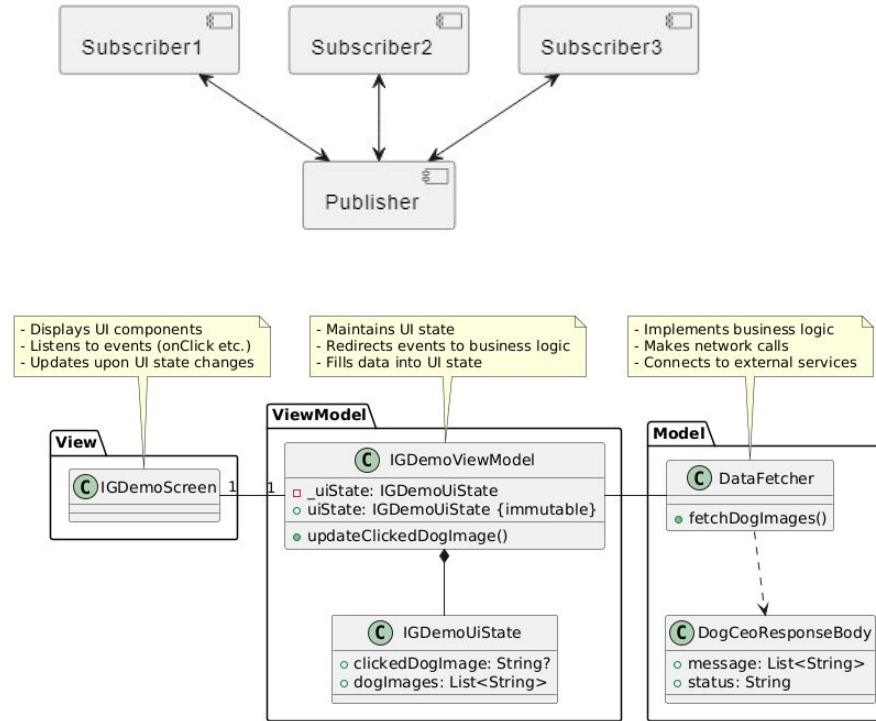
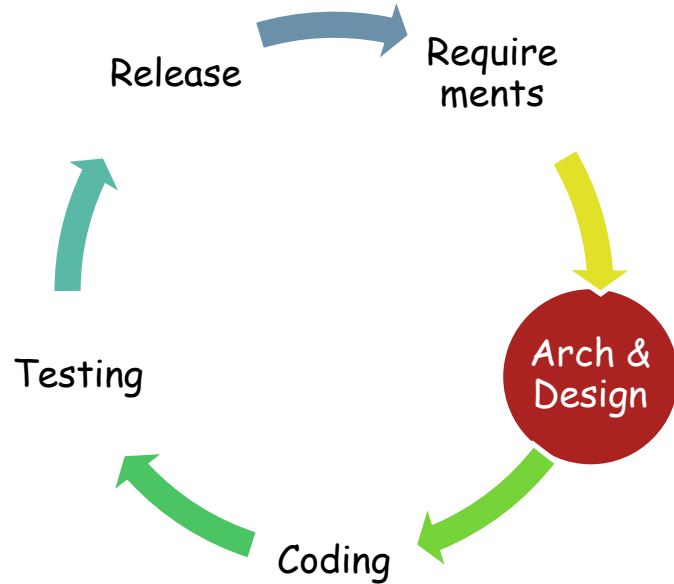
Overview of Syllabus: Modeling & Decomposition



Software Modeling, UML Diagrams
Workshop: Interviews with Stakeholders
A1 UML Practice
Decomposition Principles, Architectural Views
A2 Decomposition Practice



Overview of Syllabus: Architectural Styles

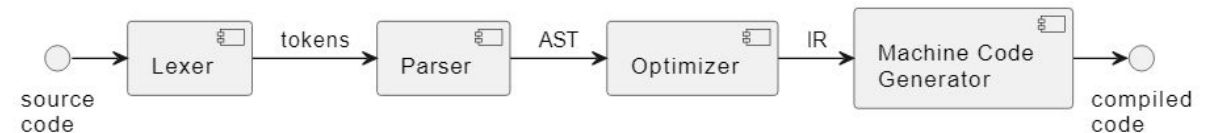


Architectural Styles 1

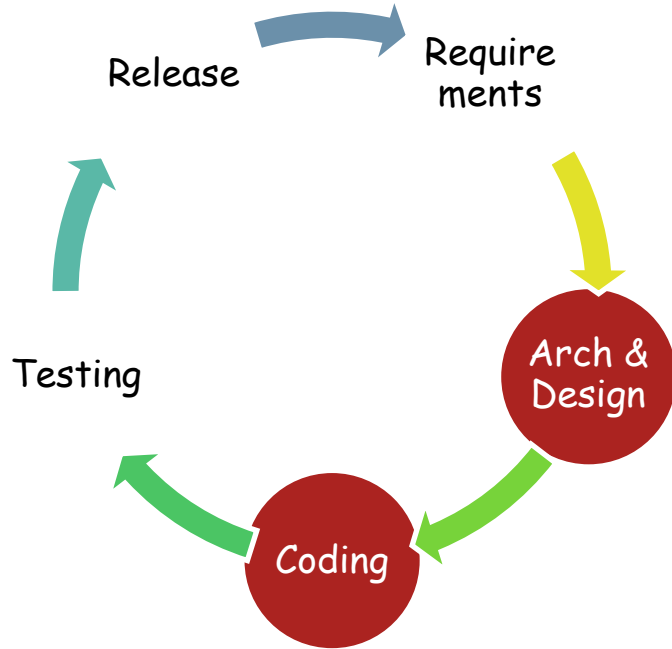
Architectural Styles 2

Architectural Styles 3

A3 Architectural Style Example



Overview of Syllabus: Design Patterns

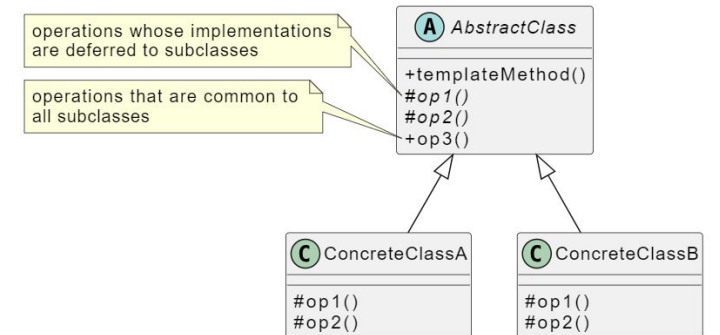
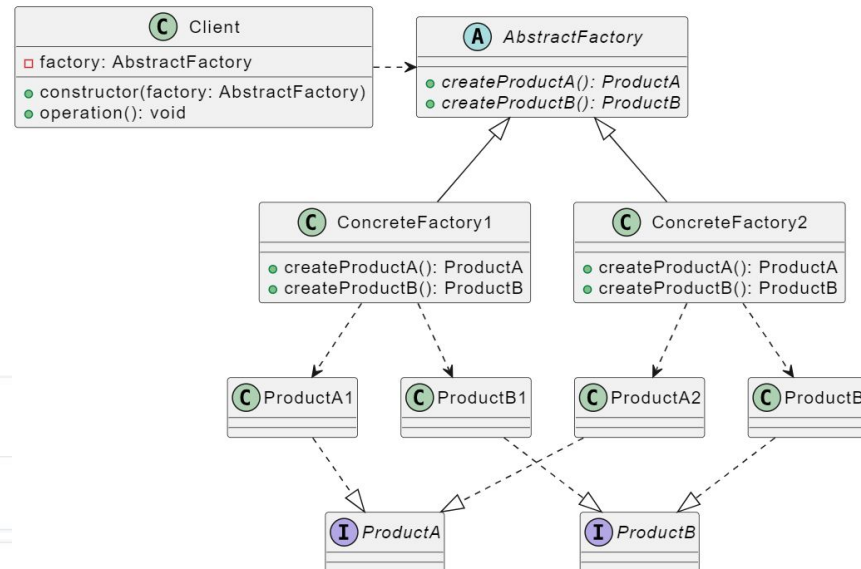
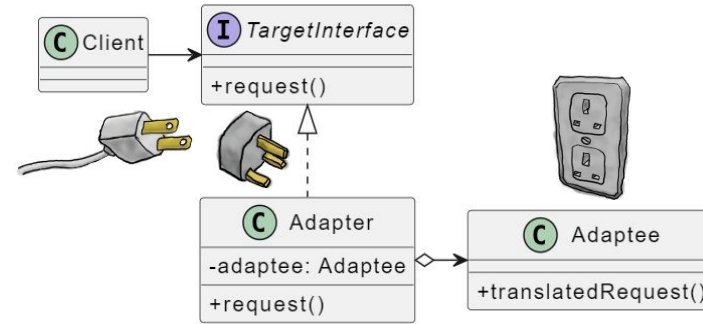


Design Patterns 1

Design Patterns 2

Design Patterns 3

A4 Design Pattern Example



What do you hope to take away?

- Course website: <https://lanorth.github.io/CS446-S26/>
- Discuss with your classmates (~10min):
 - What topics do you hope to cover?
 - What skills do you hope to develop?
- Note down your ideas
- We will discuss your ideas when we reconvene

Intended Learning Outcomes

By the end of the course, you should be able to:

- **Differentiate** how various architectural styles and design patterns *enhance* or *degrade* a system's functional and non-functional properties
- **Generate** and **justify** an architecture and/or design given a collection of requirements
- **Produce** and **present** *concise* and *unambiguous* architecture and design descriptions
- **Create** and **implement** an architecture and design, *refining* it into a complete system

Intended Learning Outcomes - Project

This is a project-based learning course. The main takeaways through the course project will be:

- The mobile app that you build along with its associated artifacts (e.g., architecture and design documents)
- Experience to work in a team to build a software app from the ground up
- An opportunity to practice pitching a software idea and presenting/defending what you built
- ***HAVE FUN!***

Assessments

- Refer to details on course website
- Determined:
 - milestones
 - grade distributions
 - details of P0, P1, grad project
- TBD:
 - The details of each assessment item will be released at least two weeks before its due date

For undergrad students:

Assessment Type	Value
Project (Team)	50%
Assignments (Individual)	20%
Final Exam (Individual)	30%

For grad students:

Assessment Type	Value
Project (Team)	50%
Assignments (Individual)	20%
Grad Project (Individual)	20%
Final Exam (Individual)	10%

Project

- You will build an Android app in teams of six
- Estimated developing effort:
6 team members * 6h/week * ~10 weeks = ~360 hours in total
- Goals
 - to create ***something useful***;
 - to learn about architecture styles and design patterns through application in a development setting;
 - to leverage current software development technologies;
 - most importantly, to have fun along the way!
- -> course website: milestones, restrictions, examples

Previous Project Showcase

https://www.youtube.com/playlist?list=PLUFKiilfEaM0u847BM1-q_s1pGBQJdAwpC

<https://www.youtube.com/playlist?list=PLUFKiilfEaM1AmOlaylupTPED7kVjrsNt>

What could *you* build?

- think of an area of your life that represents a ***pain point***
 - finding a place to eat that fits your group budget, transportation and dietary requirements
 - determining if it's safe to go to class during cold and flu season
- or, think of an under-explored/served market
- or, something that simply interests you

What could *you* build?

- it's generally not recommended for projects to:
 - involve complex server infrastructures
 - because these can be exceptionally **challenging** to debug
 - be games
 - games are notoriously complex involving both art and technical components
- however for S26 ...
 - ***I will allow games*** ...
 - you can use Unity, Unreal, or Godot engines
 - so long as the other project requirements are met

Academic Integrity

- You **collaborate** (with teammates and classmates) to complete your project
 - DO: ask questions on Piazza and answer other students' questions
 - DO: talk to other teams if you want
 - DO: use online resources (e.g., StackOverflow) and Gen AI (e.g., ChatGPT) to help you learn, code, etc.
- However, collaboration $\neq$ **plagiarism/cheating**
 - DO NOT: claim someone else's work (including Gen AI) as yours
 - DO: add citations/acknowledgements when you receive help from other teams, online resources, **Gen AI**, etc.

Generative AI

You can use Generative AI, *if attributed*, for ...

1. Graphic elements including
 - a. icons, logos, images, backgrounds
 - b. 3D models, animations, video clips (as appropriate)
2. Audio elements including
 - a. sound effects
 - b. background music
 - c. narration, dialogue
3. Text elements including
 - a. proofreading/copy editing
 - b. storyline dialogue/generation

Action Items

- Review course website
- Review requirements of P0 and P1
- Start looking for teammates
- Start installing Android Studio (may take some time)

Welcome to the class! I hope you enjoy it!