Software Design & Architecture

Introduction to Course

Agenda

- Logistics
- Overview of the syllabus
- Team project

Winter 2026. Instructor: Pengyu Nie.
Acknowledgements: slides adapted from previous versions by Mei Nagappan and Shane McIntosh, which are adapted from previous versions by Zhen Ming Jiang, Ahmed E. Hassan, Reid Holmes.

Your Instructor

- Pengyu Nie
- Assistant Professor @ UWaterloo CS
  joined not long ago (2023)
  - before that: PhD @ UT Austin ECE (Austin, Texas, USA)
  - before that: BSc @ USTC Physics (Hefei, Anhui, China)
- 3rd time teaching this course
- Do research on AI+SE

Your TAs

- Saarang Agarwal <saarang@>
- Liliana Hotsko <lhotsko@>
- Bihui Jin <bihui@>
- Larry Li <yinxi.li@>
- 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

Course Website

https://pengyunie.github.io/cs446-1261/

- syllabus
- lecture slides and notes
- assessment requirements

Learn

- announcements
- project deliverable / assignment submission
- grade release

Piazza

- Q&A
- Find teammates

Dates and Times

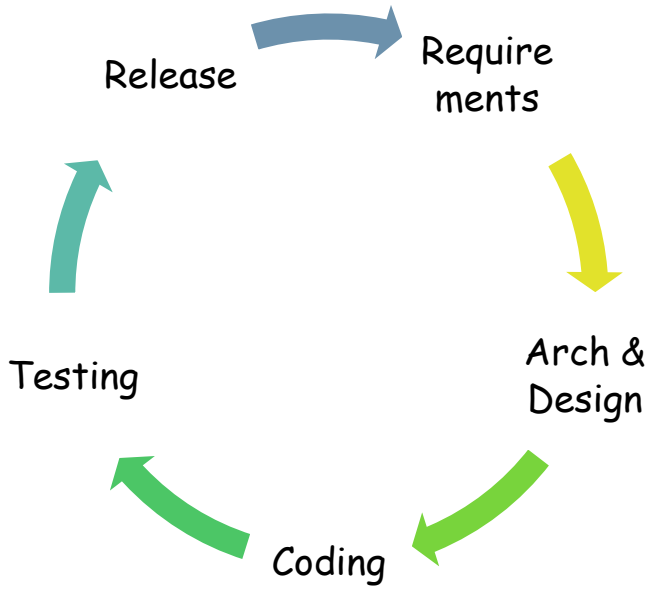
- Sec 001: MW 10:00am-11:20am @ MC 4021
- Sec 002: MW 4:00pm-5:20pm @ MC 2065
- Check the syllabus for project deadlines
  - All deadlines refer to 11:59pm Eastern Time on that day
  - Presentations are in class
- Announcements via Learn (archived) and emails (as long as you didn't filter emails from 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

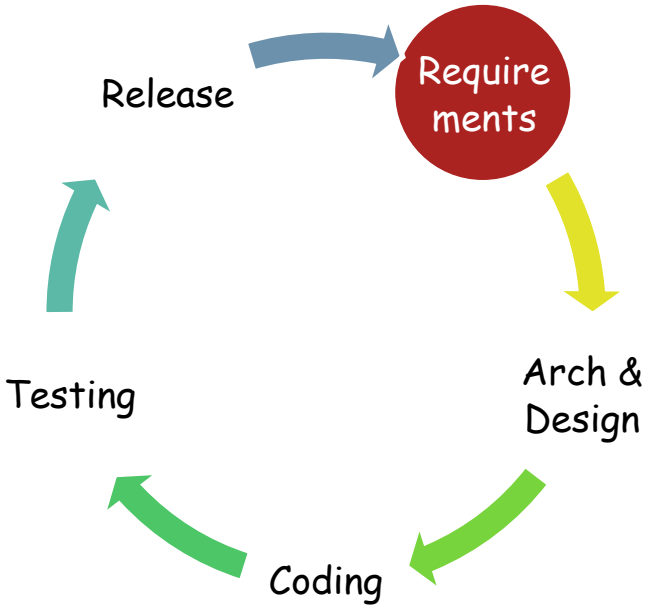
Week	Date	Lecture / Deadline
1	Jan 05 Mon	Intro to Course
	Jan 07 Wed	Intro to Software Architecture, Stakeholders
2	Jan 12 Mon	Non-Functional Requirements
	Jan 14 Wed	User Scenarios, Use Cases, Human Values
3	Jan 19 Mon	Software Modeling, UML Diagrams
	Jan 21 Wed	Workshop: Interviews with Stakeholders
4	Jan 26 Mon	Decomposition Principles, Architectural Views
	Jan 28 Wed	Architectural Styles 1
5	Feb 02 Mon	Architectural Styles 2
	Feb 04 Wed	Workshop: Android Studio, Kotlin
6	Feb 09 Mon	Architectural Styles 3
	Feb 11 Wed	Workshop: Jetpack Compose
9	Mar 02 Mon	Design Patterns 1
	Mar 04 Wed	Workshop: Testing
10	Mar 09 Mon	Design Patterns 2
	Mar 11 Wed	Workshop: Continuous Integration
11	Mar 16 Mon	Design Patterns 3
	Mar 18 Wed	Workshop: Design Assessment and Verification
12	Mar 23 Mon	Evolution of Software Design and Architecture
	Mar 25 Wed	Workshop: Release Engineering



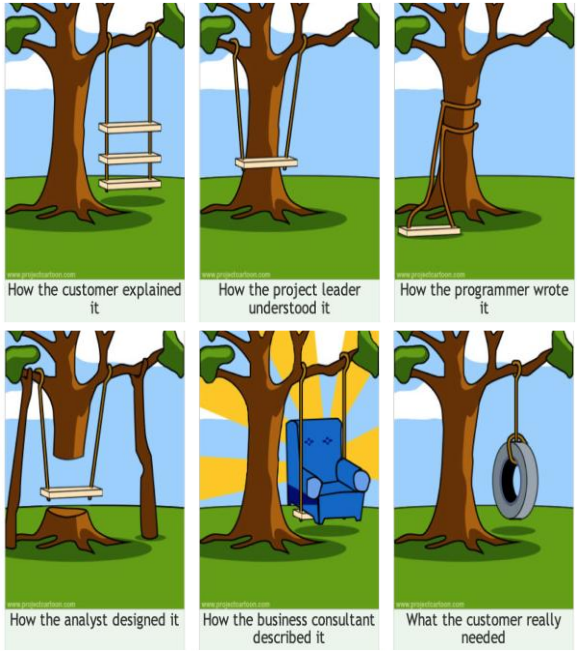
Course Delivery

- 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

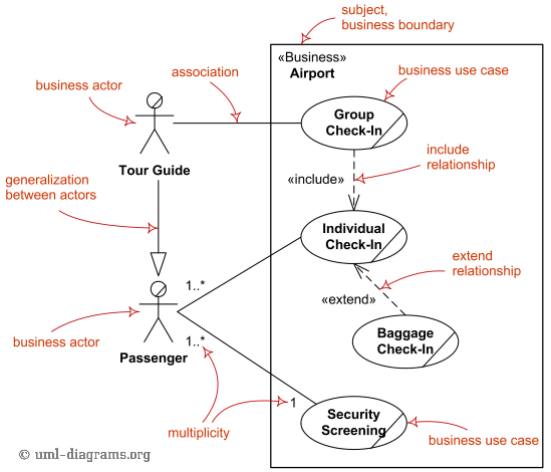
Overview of Syllabus: Requirement Phase



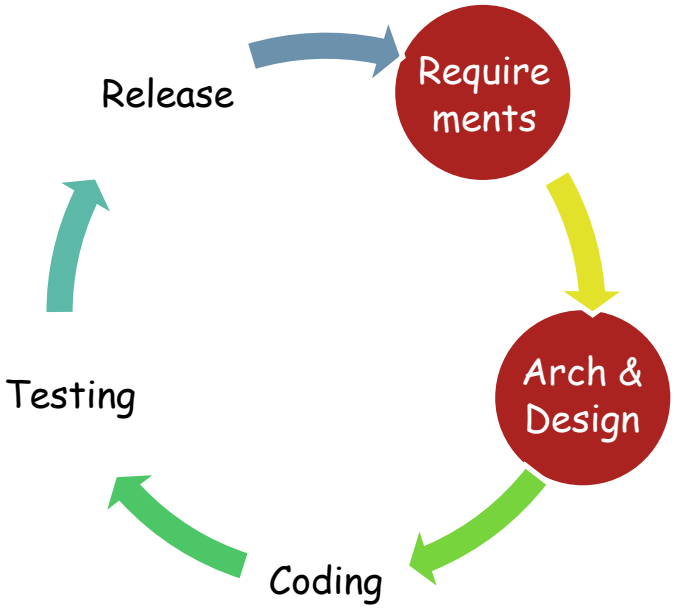
	Jan 07 Wed	Intro to Software Architecture, Stakeholders
2	Jan 12 Mon	Non-Functional Requirements
	Jan 14 Wed	User Scenarios, Use Cases, Human Values
	Jan 21 Wed	Workshop: Interviews with Stakeholders



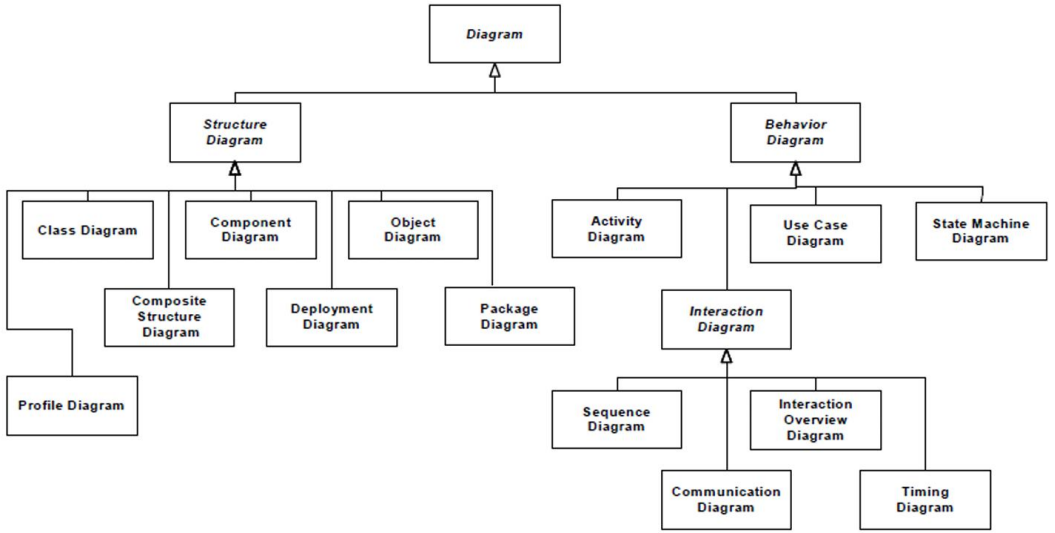
Efficiency	Privacy	Availability	Complexity
Usability	Security	Reliability	Readability
Portability	Survivability	Robustness	Heterogeneity
Accessibility	Safety	Fault-tolerance	
		Scalability	
		Evovability	

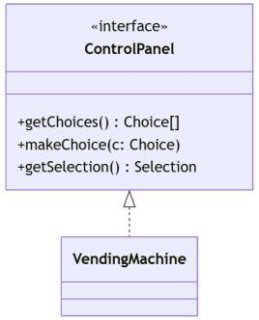


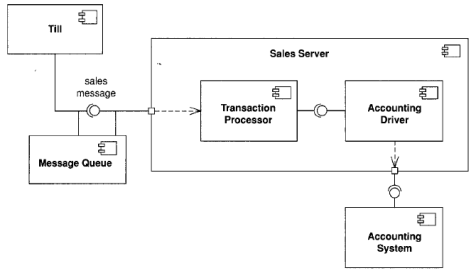
Overview of Syllabus: Modeling & Decomposition

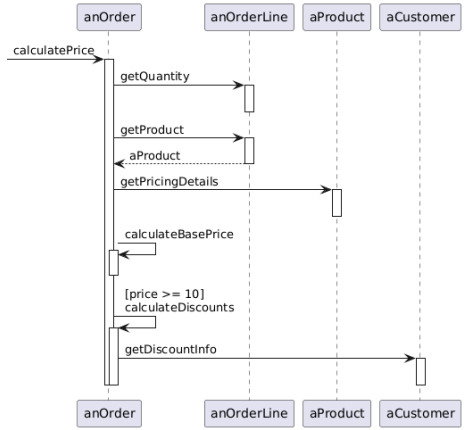


3	Jan 19 Mon	Software Modeling, UML Diagrams
	Jan 21 Wed	Workshop: Interviews with Stakeholders
	Jan 23 Fri	**A1** UML Practice
4	Jan 26 Mon	Decomposition Principles, Architectural Views
	Feb 06 Fri	**A2** Decomposition Practice

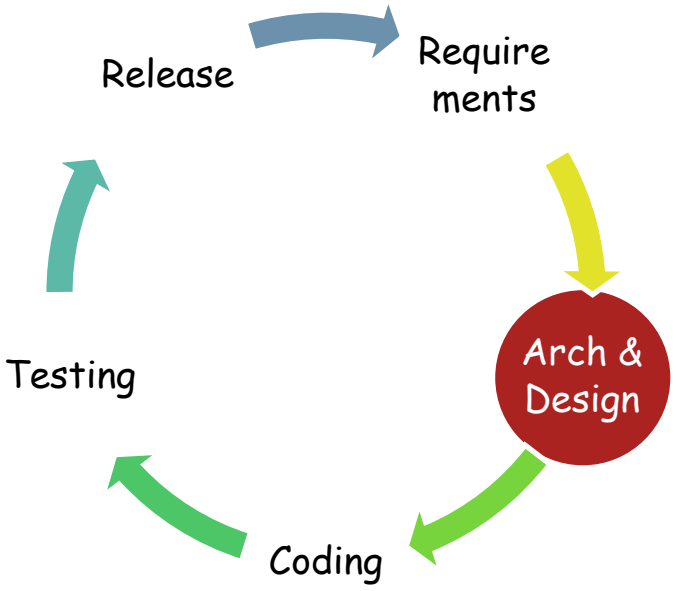


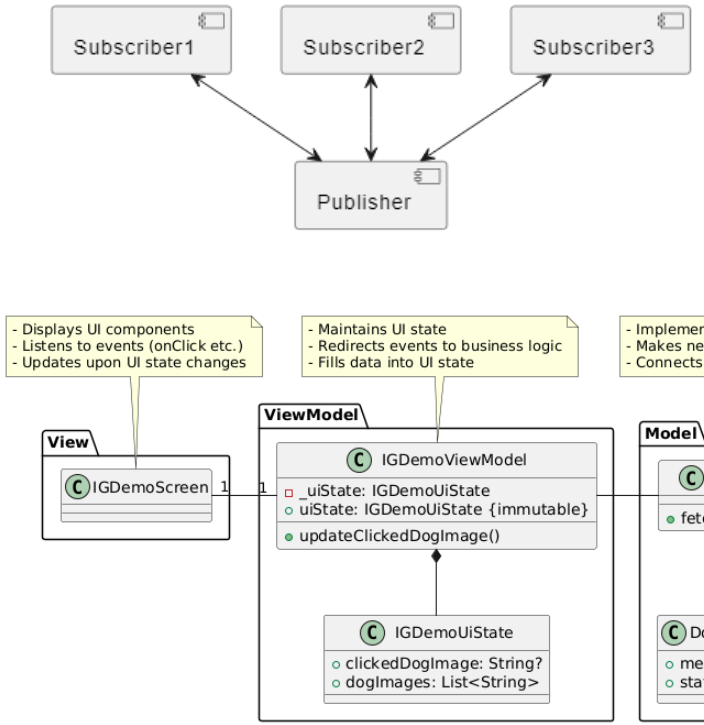


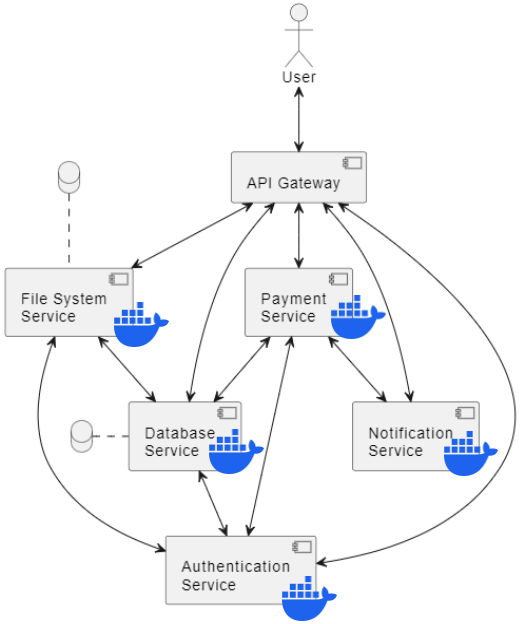


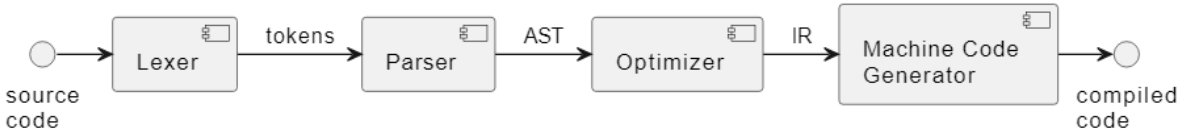


Overview of Syllabus: Architectural Styles



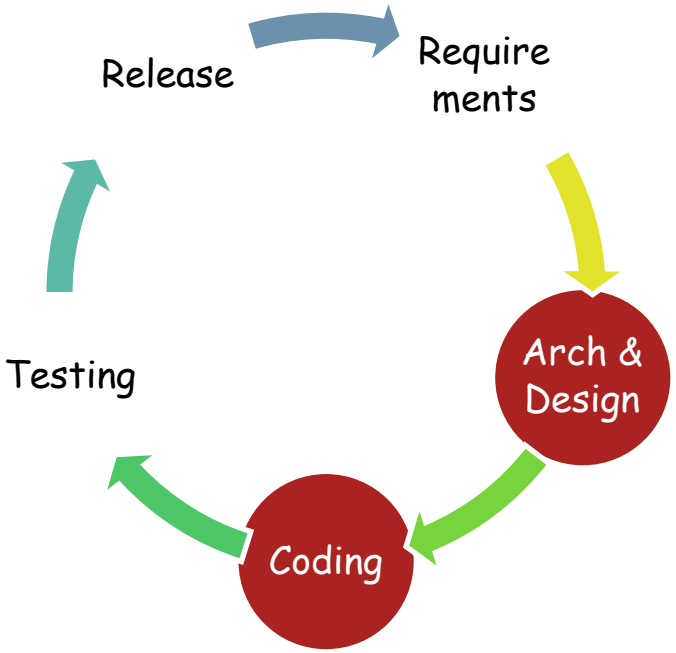




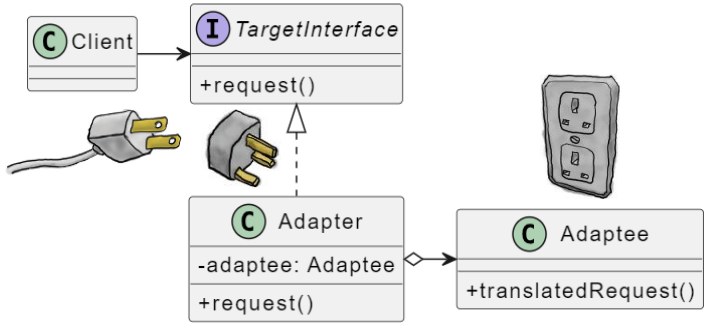


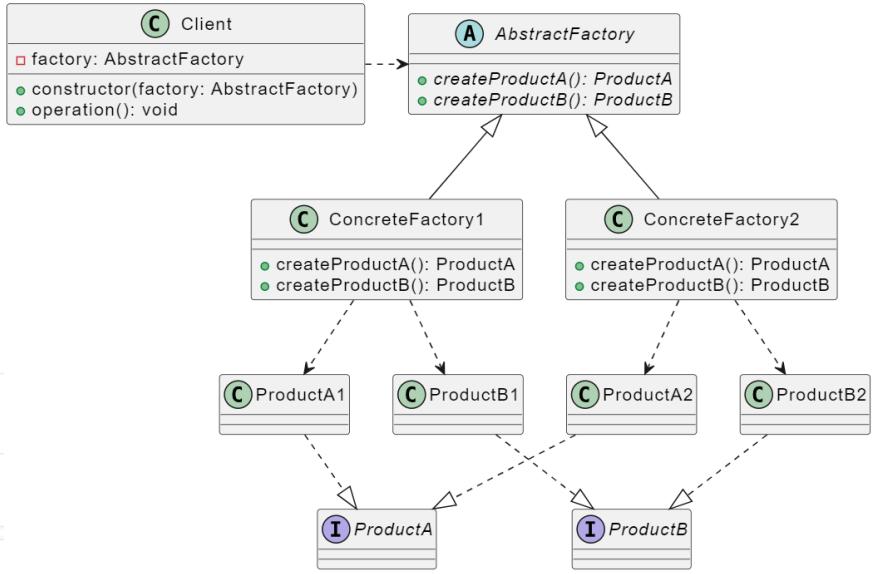
	Jan 28 Wed	Architectural Styles 1
5	Feb 02 Mon	Architectural Styles 2
6	Feb 09 Mon	Architectural Styles 3
	Feb 27 Fri	**A3** Architectural Style Example

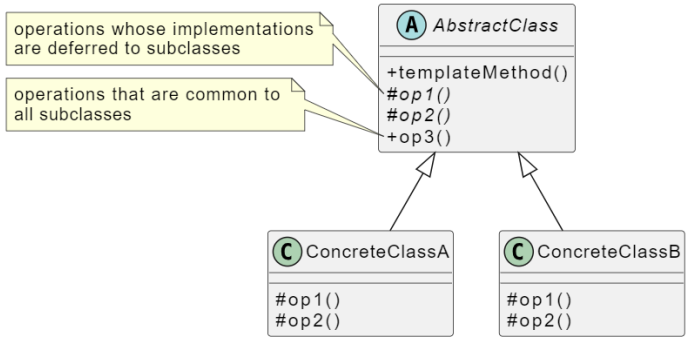
Overview of Syllabus: Design Patterns



9	Mar 02 Mon	Design Patterns 1
10	Mar 09 Mon	Design Patterns 2
11	Mar 16 Mon	Design Patterns 3
	Mar 20 Fri	**A4** Design Pattern Example



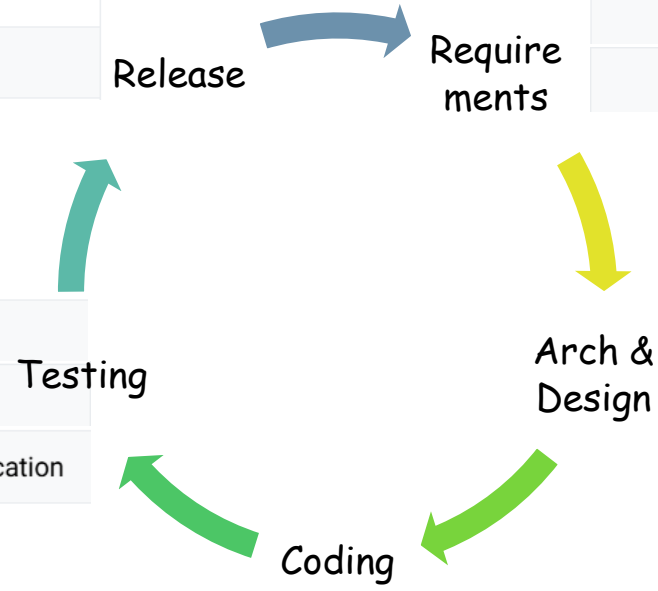




Overview of Syllabus: Project & Workshop

	Mar 25 Wed	Workshop: Release Engineering
13	Mar 30 Mon	**P5** Final Presentation Day 1
	Apr 01 Wed	**P5** Final Presentation Day 2
14	Apr 06 Mon	**P6** Final Report

	Jan 16 Fri	**P0** Team Formation
	Jan 21 Wed	Workshop: Interviews with Stakeholders
	Jan 23 Fri	**P1** Project Setup
	Jan 30 Fri	**P3** Project Proposal



(covered by the lectures)

	Mar 04 Wed	Workshop: Testing
	Mar 11 Wed	Workshop: Continuous Integration
	Mar 18 Wed	Workshop: Design Assessment and Verification

	Feb 04 Wed	Workshop: Android Studio, Kotlin
	Feb 11 Wed	Workshop: Jetpack Compose
8	Feb 23 Mon	**P4** Prototype Presentation Day 1
	Feb 25 Wed	**P4** Prototype Presentation Day 2

What do you hope to take away?

- Course website: https://pengyunie.github.io/cs446-1261/
- 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 atifacts (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

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

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 ≠ plaglarism/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.

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!