Software Engineering Design

Fall 2025

Prof. Jonathan Lee (李允中)
Department of Computer Science and Information Engineering
National Taiwan University

CSIE 5734

- Instructor: Prof. Jonathan Lee (李允中)
- Email: jlee@csie.ntu.edu.tw

- Class Hours: Tuesday 08:10 am – 11:10 am
- Online Synchronous Learning on Google Meet

- Office Hours: by Emails

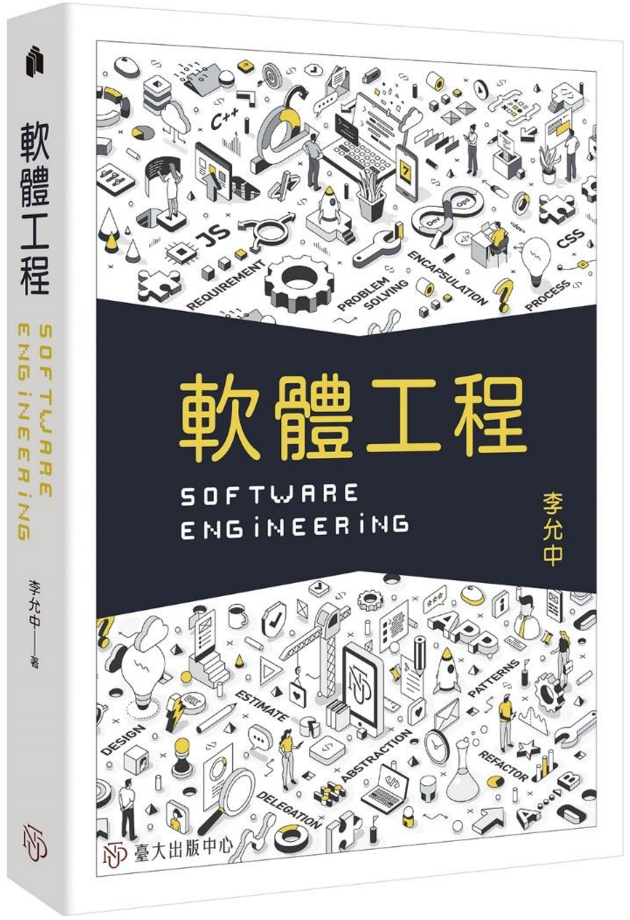
Grading

- (35%) Class attendance and participation (10%), individual baseline quiz (0%), individual object-oriented modeling quiz (10%), and weekly team homework (15%).
- (25%) Team project: Open Source Walk-through (6000 LOC)
  - Spring Framework
  - Presentation, Work Breakdown Structure (WBS), Meeting Minutes, Class Diagram, and Design Patterns.
- (15%) Midterm Exam (Individual)
- (25%) Final Exam (Individual)

Course Materials & Reference

- Course materials and slides at URL:
  - https://cool.ntu.edu.tw/courses/52011
- "Design Patterns: Elements of Reusable Object-Oriented Software," Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley Professional, 1995.
- "Head First Design Patterns," Eric Freeman, Elisabeth Freeman, Kathy Sierra, and Bert Bates, O'Reilly Media, 2004.
- "UML 2.0 in a Nutshell," Dan Pilone and Neil Pitman, O'Reilly Media, 2005.

Course Materials & Reference

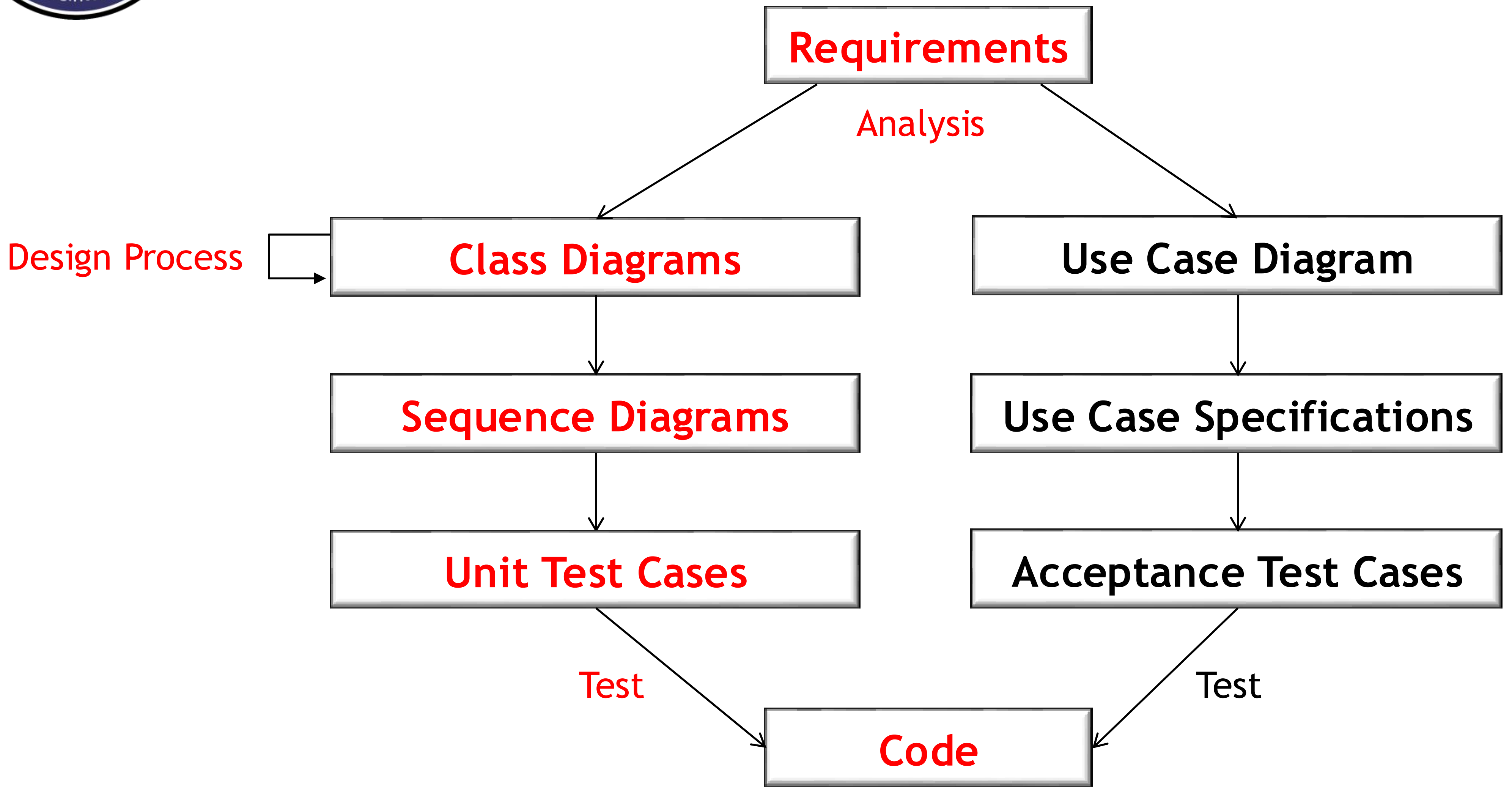


- 書　　名：軟體工程 (Software Engineering)
- 作　　者：李允中
- 出版日期：2024年10月出版
- 出版單位：國立臺灣大學出版中心
- 紙 本 書：三民書局 / 五南網路書店 / 博客來 / 國家書店 / 灰熊愛讀書 / 誠品網路書店 / 讀冊生活 /
- ISBN：978-986-350-864-9
- GPN：1011301191

Course Outline

Week 1 (9/2)	Syllabus; Homework (1)
Week 2 (9/9)	Quiz: Baseline (3 hours; from 8:10am to 11:10am); Homework (2)
Week 3 (9/16)	Project Management (WBS); Homework (3)
Week 4 (9/23)	Object-Oriented Concepts; Homework (4 and 5)
Week 5 (9/30)	Object-Oriented Modeling Practice
Week 6 (10/7)	Quiz: Object-Oriented Modeling (3 hours)
Week 7 (10/14)	Design Patterns Concepts (Homework 7-0,1,2)
Week 8 (10/21)	Design Patterns (I) (Homework 8-0,1,2)
Week 9 (10/28)	Midterm Exam (5 hours, from 7:10 am to 12:10 pm)
Week 10 (11/4)	Design Patterns (II) (Homework 9-1,2,3)
Week 11 (11/11)	Design Patterns (III) (Homework 9-2.1, 9-3.1, 10-1,2,3,4)
Week 12 (11/18)	Design Patterns (IV) (Homework 11-1,2,3,4)
Week 13 (11/25)	Design Patterns (V)
Week 14 (12/2)	Team project presentation (I)
Week 15 (12/9)	Team project presentation (II)
Week 16 (12/16)	Final Exam (5 hours, from 7:10 am to 12:10 pm)

Software Design vs Software Engineering



Software Design

- Start with problem statements
- Model with class diagrams
- Refactor with a design process involving the use of:
  - object-oriented concepts
  - design principles
  - design patterns

Object-Oriented Concepts

- Inheritance
- Polymorphism
- Dispatching

- Encapsulation
- Abstraction
- Composition
- Delegation

OO Design Principles

- Inherit the most important features and delegate the rest
- Encapsulate what varies
- Favor composition over inheritance
- Program to interface, not implementation
- Strive for loosely coupled designs between objects that interact
- Classes should be open for extension but closed for modification
- Depend on abstractions. Do not depend on concrete classes
- Only interact with close classes
- ………

Design Patterns (GoF)

- Creational: Involve object creation
  - Factory Method, Abstract Factory, Builder, Prototype, and **Singleton**.
- Structural: Compose classes or objects into larger structures
  - Adapter, Bridge, Composite, Decorator, **Façade**, Flyweight, and Proxy.
- Behavioral: Concern with how classes and objects interact and distribute responsibility
  - Interpreter, Template Method, Chain of Responsibility, Command, Iterator, Mediator, Memento, Observer, State, **Strategy**, and Visitor.

"Design Patterns: Elements of Reusable Object-Oriented Software," Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley Professional, 1995

Enroll in my Class for All Students

- Send me an email, which is required to enroll in the course, with the following information about yourself.
  1. Indicate whether you have enrolled in my class or not.
  2. Your photo, name, undergraduate and graduate major(s), and the university you received your bachelor's degree from.
  3. Can you program in Java (prerequisite)?
  4. Where did you learn about this course, including the names of your friends or co-workers?
  5. Why do you want to take this course?
  6. What do you expect to take away from this course?

- After I review all the emails, I will send acceptance email notifications to those who still need to enroll in the class.

- No sit-in.

Next Week (9/9): Baseline Quiz

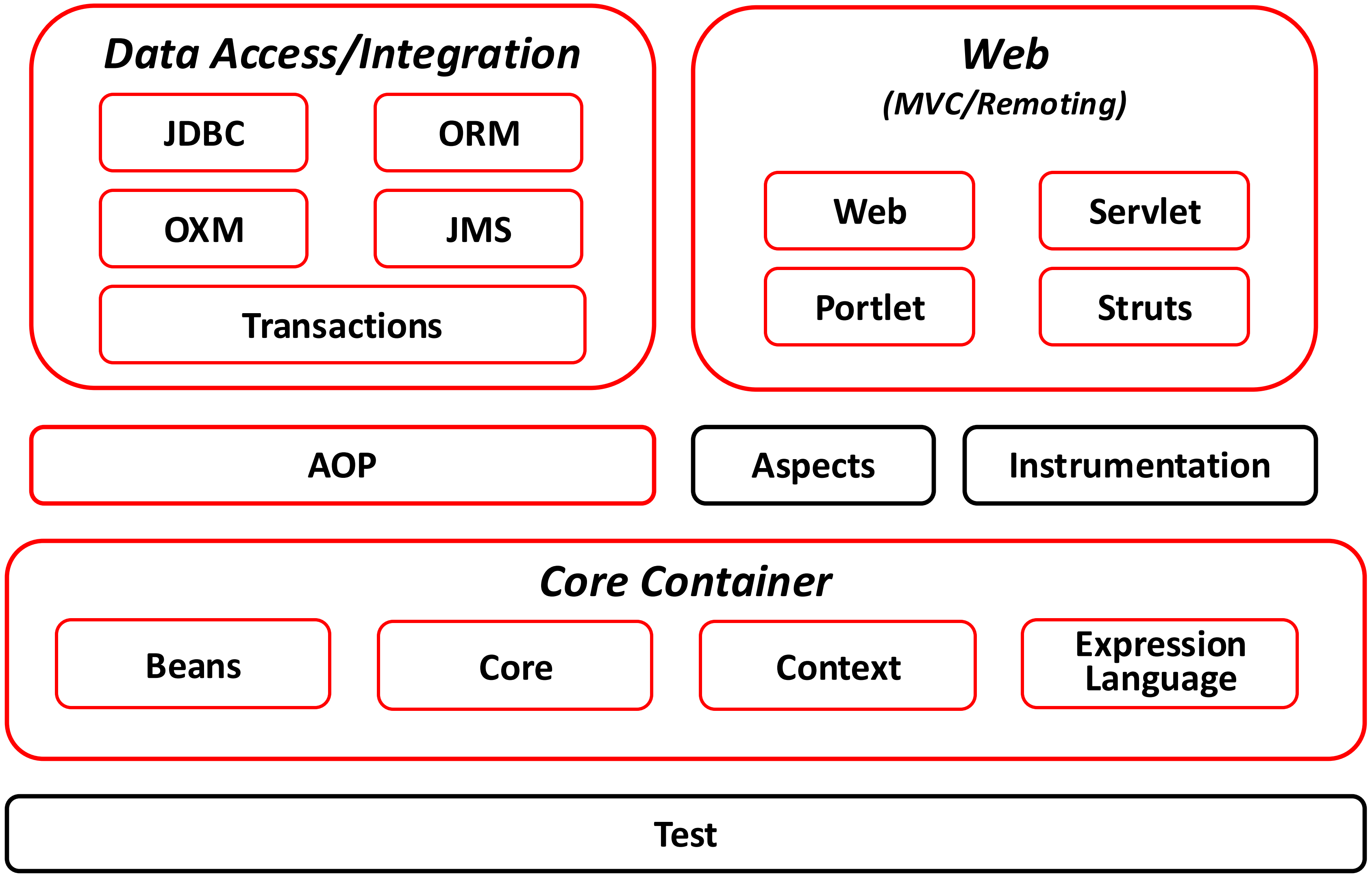
- **Online quiz** (individual)
- Programming Language: Java only
- Time: 8:10 – 11:10

- Grade: Only serves as a baseline and does not count in the final grade
- **Taking the baseline quiz is required for those who want to enroll in my class.**

Homework (1)

- Submit a list of your team members after the quiz (maximum four members for each team) (due 9/9).
- Team Project: Open-source code walk-through (6000 lines of code or 6KLOC, for each team member, 500 LoC/week) and develop its class diagram design every week.
- Present your choice of the part of Spring Framework, Spring Boot, or Spring Cloud for the team project (due 9/16).

Spring Framework Architecture



Essence of Software Design

- Always delegate to others to do the work if possible.
- Have limited or no knowledge of the classes that are delegated to.
- Have limited or no knowledge of how the delegated implements the requests.

重要提醒

- 課堂回饋：
  - 如果可以的話，會希望這門課開在下午或晚一點開始。(若期中/期末要考比較久，可早點開始)因為在早上頭腦昏沈的情況下，有些原本很簡單的東西，難度都會不知怎地Level up (像Factory- method和Abstract- Factory，我在準備期末時覺得他們都滿簡單的，但上課時看到畫很多線的class diagram+頭腦尚未開機，就對它產生一股排斥感)。
  - 早八好累。。。一開始還沒教design pattern的時候，每週code tracing常常只是就畫 class diagram而已，然後理解class在做什麼，感覺只是看code畫圖這樣。比如類似我看的View Subsystem，只是在implement各種view，做法都非常相似，變得會有些枯 燥。當學了許多design pattern之後，才會慢慢理解其中所含有的一些design pattern，也慢慢理解subsystem的設計。我覺得有收穫的一點是透過上課認識了許多design pattern以及在spring上看到類似的設計，這樣感覺挺不錯的。
  - 小組報告的形式，這邊沒有要抱怨組員的意思，不過其實整個每週meeting的形式跟我對於整個團隊合作想像的有點差距。學期初的功課數量不會太多時還會是大家一起討論跟一起做功課的形式，後來漸漸變成大家一人做一項，然後meeting報告的方式，就比較沒有具體的參與到每個作業中了。而且期末報告我最初以為是會有一個全組的issue track或針對整個framework的功能或pattern做報告，結果後來變得比較像是個人報告自己的部分。整體說是teamwork更感覺像是讀書會的形式。
- 請同學在做作業時，務必與組員進行討論與合作。切勿僅由一位同學獨立完成報告，並且在沒有進行充分討論的情況下就結束作業。
- 本學期課程中不得使用LLM和Copilot，本學期課程考試不得使用LLM和Copilot及禁止上傳老師的課程講義到LLM和Copilot。