



# Software Design

## Fall 2024

Shin-Jie Lee (李信杰)

Associate Professor

Computer and Network Center

Department of Computer Science and Information  
Engineering

National Cheng Kung University



# 人力市場需求

(122-1)【軟體研發中心專區】- 軟體工程師 05/23更新

## 本公司其他工作

工作技能 軟體工程系統開發、軟體工程開發、軟體程式設計、軟體程式設計

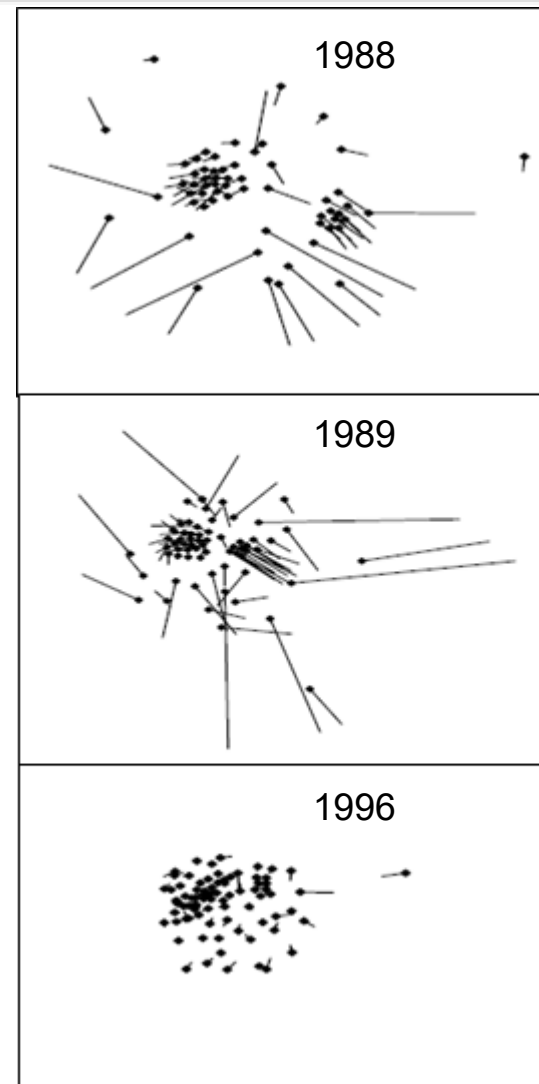
其他條件 必要條件：

1. 熟悉 C/C++ 開發，具備 OOD 與模組化開發概念。
2. 具有 UML 概念與使用經驗。
3. 熟悉 Design Pattern，並有盡力維護軟體編碼品質習慣。
4. 熟悉 Docker 虛擬容器開發。
5. 熟悉 git 與 gitlab 或相關版本控制工具的使用，了解 CI/CD 流程。
6. 熟悉於 Linux 環境進行開發。



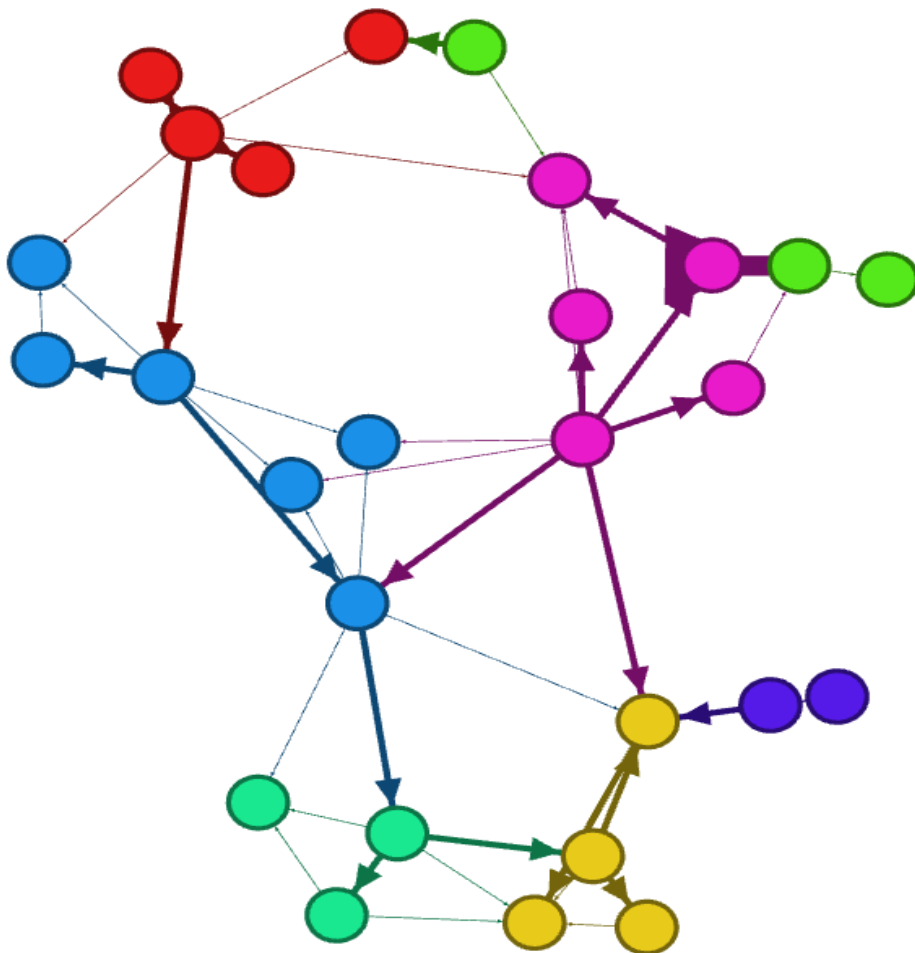
# Software/Code Decay

- ❑ A unit of code (in most cases, a module) is decayed if it is *harder to change than it should be*, measured in terms of effort, interval and quality.
- ❑ An Example: The Telephone Switches Project
  - Fifteen-year old real-time software system for telephone switches
  - 100,000,000 lines of source code (C/C++) and 100,000,000 lines of header and make files, organized into some 50 major subsystems and 5,000 modules.
  - More than 10,000 software developers have participated.
  - A module within one of the clusters is often changed together with other modules in the cluster but not with other modules
    - Head of each tadpole-like shape corresponds to a module



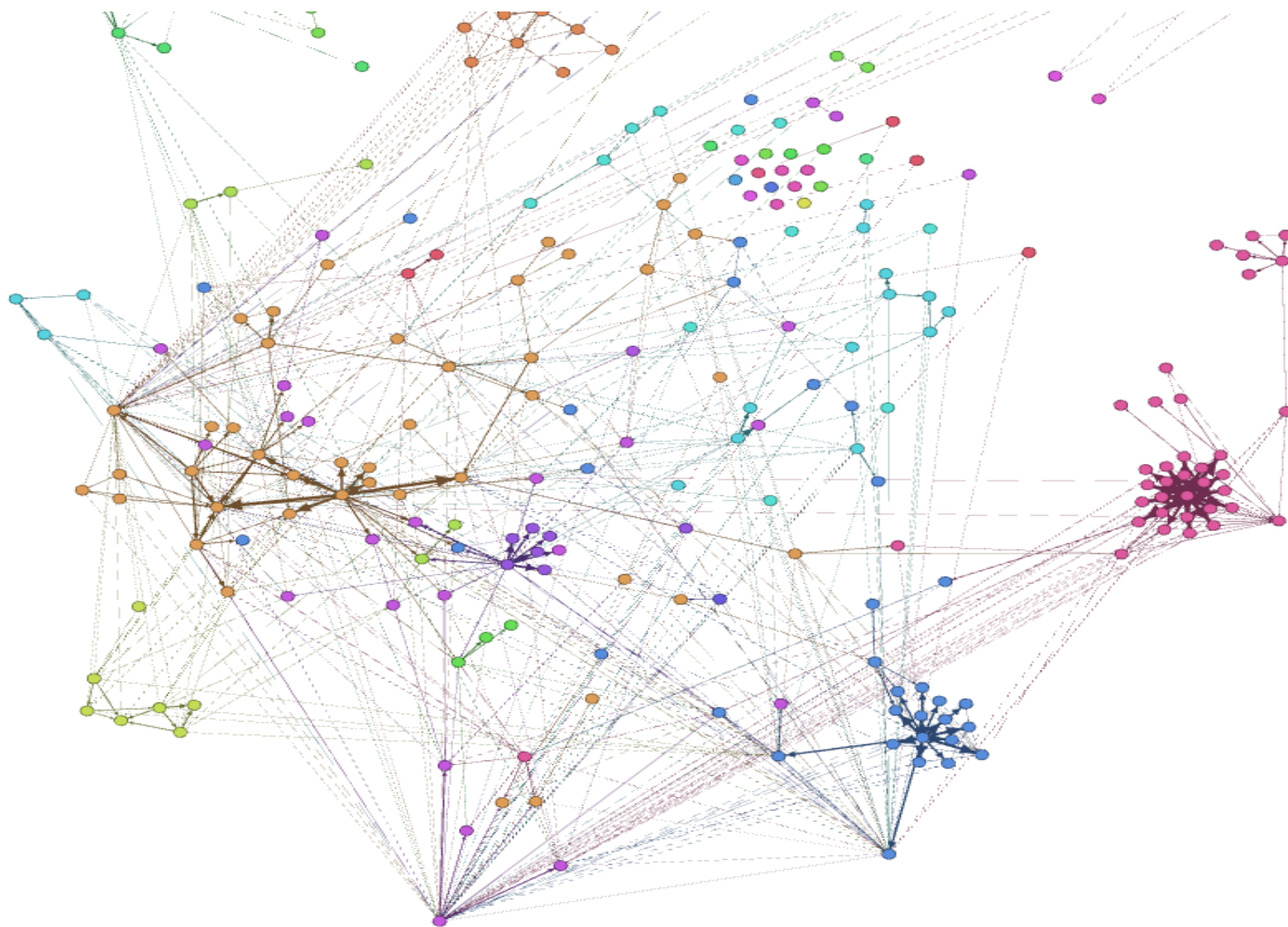


# Code Structure





# Code Structure





# Software Design @ Software Engineering

## Software Lifecycle

Waterfall, Agile Method, DevOps

## OO Design and Programming

**Object-Oriented Concept, UML, Design Smell, Design Principle, Design Patterns, Coding, Refactoring, Unit Testing, Debugging**



# Software design is challenging

**TABLE 4.** The number of replies given for every choice in the question “How challenging were the following when you have moved from simple projects to relatively larger ones?” The answer represent challenge levels, 1 being the easiest and 5 the most challenging. Level 0 represents no experience.

Tasks	0	1	2	3	4	5
Requirements	9	15	21	19	12	2
Software Design	5	13	16	17	21	6
Software Development	6	15	26	17	12	2
Software Tests	10	9	18	14	15	12
Project Management	16	15	13	18	6	10
Personal Time Management	9	18	21	14	10	6
Communication	6	30	15	11	9	7
Project tools	12	20	19	16	5	6



# 重構即設計





# 重構的基礎為

- Design Smells
- Design Principles
- Design patterns



理解它們並不難，  
難在熟練地辨識與運用它們



每個待重構的軟體都是獨特的，  
熟練度會決定重構效率



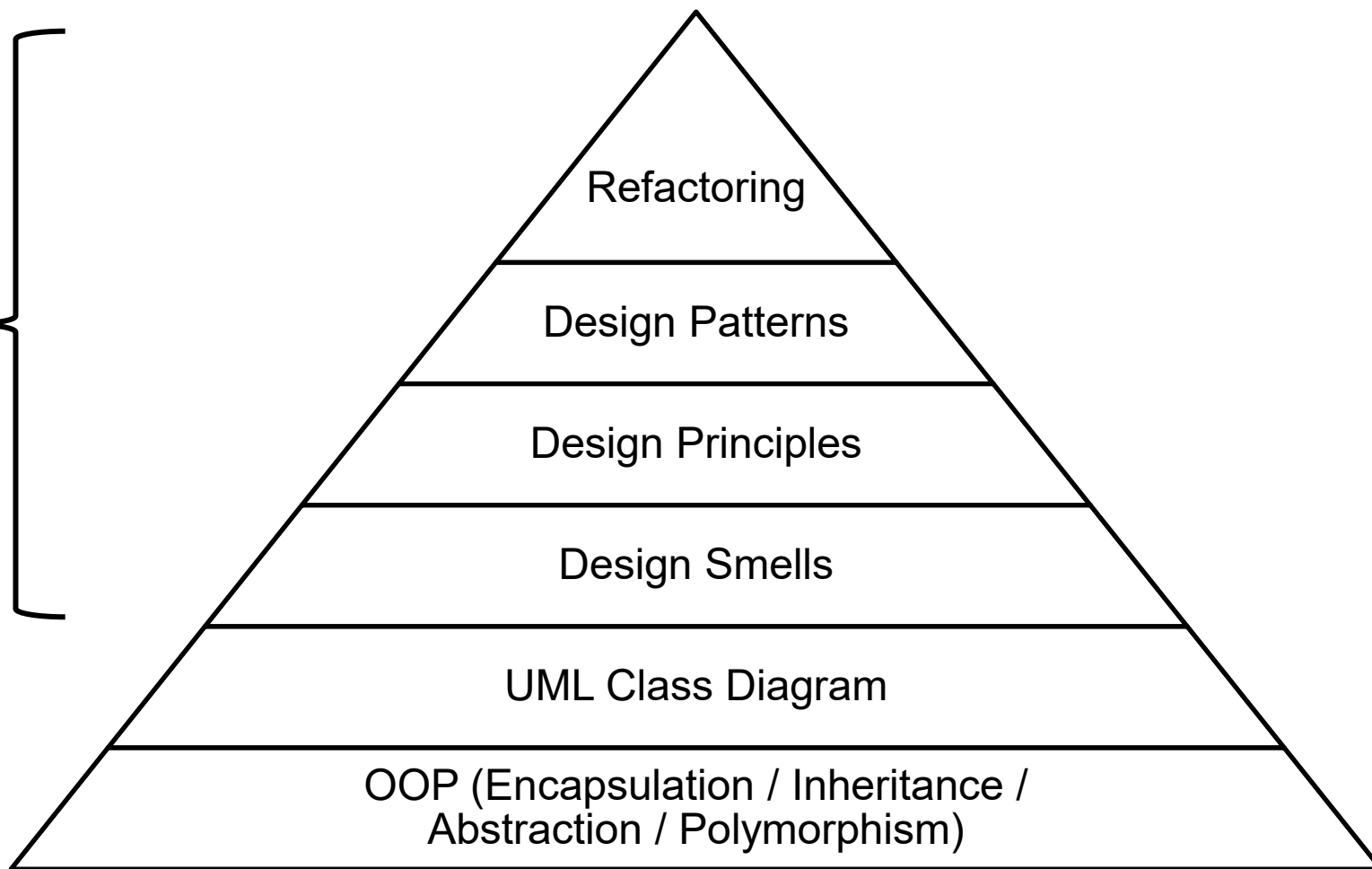
提升熟練度的方法就是  
刻意練習



# 由下而上

## Improved Design

刻意練習





# Course Outline

1	Introduction
2	Object-Oriented Concepts
3	UML Class Diagram
4	Code Smells
5	Code Smells
6	Refactoring
7	Design Principles
8	Design Principles
9	期中考
10	Design Patterns
11	Design Patterns
12	Design Patterns
13	Clean Code/Architecture
14	Clean Code/Architecture
15	業界教學
16	業界教學
17	業界教學
18	期末專題觀摩



# Grading

方法	百分比%
作業 Assignments	30
期中考 Midterm Exam	20
業界教學心得報告	25
個人專題期末報告	25



# 注意事項

- ❑ 本學期授課以Java為主，請先熟悉Java與Eclipse開發環境，以利使用ObjectAid UML外掛工具
- ❑ 請務必自帶筆電+滑鼠至教室(非平板)，以利實作練習
- ❑ 請確認筆電麥克風收音功能正常，以利課堂上利用Google Meet分享練習題
- ❑ 每週作業請準時繳交，逾期不計分





# Course Materials & Reference

- ❑ Slides at <http://moodle.ncku.edu.tw/>
- ❑ <https://refactoring.guru/>
- ❑ Refactoring Kata at <https://codingdojo.org/kata/> and <https://katalog.rocks/refactoring>
- ❑ “Refactoring: Improving the Design of Existing Code,” Martin Fowler, Addison-Wesley Professional, 2018.
- ❑ “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.
- ❑ “Clean Architecture: A Craftsman's Guide to Software Structure and Design,” Robert C. Martin, Prentice Hall, 2017.
- ❑ “Clean Code: A Handbook of Agile Software Craftsmanship,” Robert C. Martin, Prentice Hall, 2008.