Object Oriented Design Pattern
(ITP 30008) Spring, 2022

Prof. Keum-Young Sung
kysung@handong.ac.kr;
Office: NTH 308;

Office Hour : By Appointment; mainly **Tues**. **and Thursday** between **5:00 PM and 6:00 PM,**
or any other time with appointment
Class Hour: **Tuesday and Friday,**
the 3rd class period (11:30am ~ 12:45pm),
Lecture Room: **NTH 311**

Pre-requisite Knowledge: Java
Type of Class: Lecture, Program Test, **Short Quiz**, Analysis of Source Code with Design Pattern, Java Review

Course Description

- Students learn the essential concepts about the **object-oriented** paradigm with Java programming language, and **design patterns** for making **reusable** and **maintainable** software system.
- Students improve Java proficiency up to intermediate level with design pattern techniques.
- Students build the capability to apply Java language to **actual problem solving**.

Why Java?

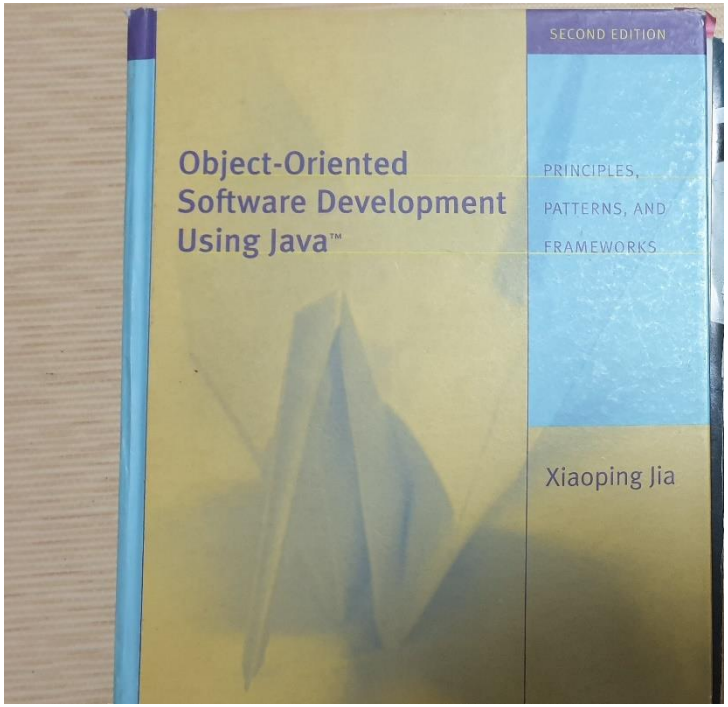
- High Degree of **Modularity** and **Reusability** and **Maintainability**
- **General Purpose** Language
- **Multithreading**
- **Distributed Computation with RMI**
- **Architecture Neutral and Platform Independence**
- **Web Server Programming with Servlet and JSP**
- **Android Programming**
- **Database Connection**
- **Automatic Garbage Collection and No Pointer**
- **Security with Sand Box Concept**

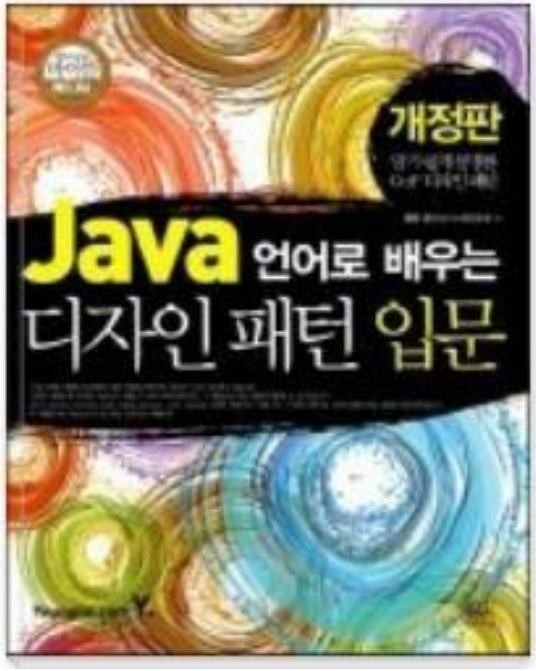
…

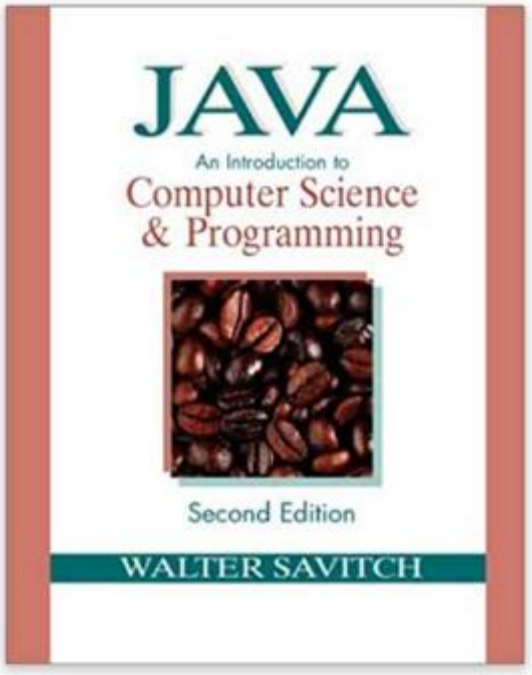
Goal

1. To provide students with a **theoretical understanding** (***inheritance, polymorphism, multithreading, abstraction, information hiding, modularity, encapsulation,*** etc) of object oriented programming techniques, and **design patterns for maintainable code**;
2. To practice **applying** object oriented techniques and **design patterns** to software development with the goal of improving **reusability and maintainability**.

Recommendable Reference Books







Recommended References

- "**Java 언어로 배우는 디자인 패턴 입문**," 유키 히로시, 개정판, 영진닷컴, 2008 **(Strongly Recommended to Read)**
- "**Object-Oriented Software Development Using Java**," Xiaoping Jia, Second Edition, Addison Wesley, 2003. (The textbook is very good for intermediate-level programmers who are familiar with basic software engineering concepts)
- "**JAVA an Introduction to Computer Science & Programming**," Walter Savitch, 6th Edition (or **upper version**), Prentice Hall, 2011. (An excellent book to introduce Java for both sides, **application and applet**)
- **Internet materials** will be posted or provided.
- **Students may purchase reference books individually.**

Main Topics

(Design patterns introduced along with OOPL concepts)

Not Strictly Following the Given Sequence (Important patterns will be treated first.

- Use of Java **IDE** – **Eclipse**, **TextPad**;
- Review of Important Java Syntax
- **Design Pattern** (Chapter 7) – Concepts and Java Code
- **Platform-Independence**, Basic Syntax and Semantics (chapters 3, 4 of the Jia Book)
- Class and **Object**, Base and Derived **Classes** (chapters 3, 5 of Jia Book)

- **Concurrent Programming** (chapter 11 and external sources)
- **Design Pattern Concepts with Chain of Responsibility** and **Template Methods**
- **Inheritance and Interface**, Single and Multiple Inheritance (chapter 5)
- Method **Overloading/Overriding and Polymorphism** (chapter 5)
- Additional Design Pattern Techniques with **Adaptor Pattern, Factory Pattern, and Command Pattern**
- Design Patterns with **Bridge pattern, Composite pattern, Decorator Pattern, Proxy Pattern.**
- **Visitor** Pattern, **Strategy** Pattern, **Memento** Pattern, **Façade** Pattern, **Mediator** Pattern, and **State** Pattern

Sequence of topics is flexible.

Some patterns may be added or omitted.

Course Evaluation and Exams

Grading criteria may be chanced depending on class performance and changing circumstances.

- ~55% programming projects for about four programming project;
- ~35% **midterm** and **final** exams (**in class exams**, not on-line).
- ~10% quiz exams

Sincerity and Personality related to class attitude and endeavor may affect the final grade!

You have to be in classroom for taking midterm and final exams(and some quiz exam).

Main **Criteria** for Grading Source Code :

1. **Maintainability** (**Understandability**, **Modifiability**),
2. **Design** Documents, and optional Developer's Note
3. **Correctness**
4. **Originality (Do not copy external source…)**

Notice with Program Submission

- Your program structure should be explained in detail with diagrams of UML. The document should be transformed into **pdf**.
- Programs should be turned **in time**.
- Do **not include Korean characters** in the source code. Otherwise, your code will not be complied and not be considered.
- File Name: **StudentName_Proj_Name.zip**
- All source code submitted should be in **Java text format**, for example, Chain.java, Process.java, etc. Format in Eclipse or any other IDE format will **not be accepted, and not be considered**.

- All your submitted files should be run with "**java codename"** in command line or command shell, not with IDE before submission.
- If you violate the given rules for your submission especially for the source codes, your working will **not be graded**.
- Remove all the package statement before submission.

~~**package chain;**~~

```
public class TestChain {
 …
```

Notices and Request

- **Honest working**;
- **Early start** on your assignment;
- **Late working will be given penalty 30% per each day up to 2 days late**;
- Assure correct working of your programs before submission.
- Grading based on **comprehensibility**, **correctness**, and **maintainability**;
- Your code should be compiled with "**javac *.java**" and run with "**java codename**"
- The **JDK Version** should be **the latest one** as in the beginning of the semester. (version 16 or later as of 2022 Feb.)

Design Document
User's Manual with Program Output
Optional Developer's Notes

- **User's Manual with Output** and **Design Documents** to be Prepared;
- **Prepare back-up copy** of your workings;
- Four absences will result in the failure of this course;
- Be on time in class.
- Any type of dishonesties will result in failure. When you think some of your class activities turned out to be dishonorable, let me know.
- Honor code will be observed for all the class workings.