Object-Oriented Programming

Syllabus

- Teaching team:
  - Lecturer: Quản Thái Hà
  - Lab instructor: Quản Thái Hà
- Online class's code:
  - Google classroom:
  - Google meet:

- At the end of the course students should
  - be familiar with the main features and limitations of the Java language
  - be able to write a Java program to solve a well specified problem
  - understand the principles of OOP
  - be able to demonstrate good object-oriented programming skills in Java
  - be able to describe, recognise, apply and implement selected design patterns in Java
  - be familiar with common errors in Java and its associated libraries
  - understand a Java program written by someone else
  - be able to debug and test Java programs
  - understand how to read Javadoc library documentation and reuse library code

- Software project development
- Software project management: testing, version control, documentation
- Team work

Grading

- Tutorial Participation: 10%
- Homework: 20%
- Midterm Examination: 20%
- Final Examination: 50%

Week	Lectures	Details
1	Introduction	- Introduction
2	Java Basics 1	- Values, variables and types - Java data types: primitive data types, non-primitive types (reference types) - Operators, keywords - Control statements: decision-making statements, loop statements, break, continue - Methods - Naming convention
3	Java Basics 2	- Arrays - Exceptions - File processing
4	OOP Introduction	- Introduction to OOP programming
5	OOP 1: User mode	- Classes as custom types, objects vs classes, class definition, constructors, access modifiers, this keyword, static data and methods, overloading, modularity, encapsulation/data hiding, immutability
6	OOP 2: Designer mode	- Identifying classes, UML class diagrams
7	Pointers, References and Memory	- Pointers and references: reference types in Java - The call stack, the heap, iteration and recursion - Pass-by-value, pass-by-reference, pass-by-sharing

Week	Lectures	Details
8	OOP 3: Inheritance	- Inheritance (is-a), aggregation (has-a), casting
9	OOP 4: Polymorphism	- Overloading, overriding - Super keyword, final keyword - Runtime polymorphism, dynamic binding, instanceof operator
10	OOP 5: Abstraction and ADT	- Abstract class and methods, interface, abstract vs interface - ADT (Abstract Data Type)
11	Design Language Evolution	- Generics, type erasure - Lambda functions, functions as values, method references, streams - Collections
12	Design Patterns 1	- Structural Design Patterns: adapter, bridge, composite, decorator, façade, flyweight, proxy
13	Design Patterns 2	- Creational Design Patterns: factory method, abstract factory, builder, prototype, singleton
14	Design Patterns 3	- Behavioral Design Patterns: chain of responsibility, command, iterator, observer, state, strategy, template method, visitor
15	GUI	- Java Swing - JavaFX

- Lecture notes, slides given by the instructors
- Cay S. Horstmann - Big Java - Early Objects, 7e-Wiley (2019)
- Eric Freeman, Elisabeth Robson - Head First Design Patterns - Building Extensible and Maintainable Object-Oriented Software-O'Reilly Media (2020)
- References
  - Goal Kicker - Java Notes for Professionals (2018)
  - Bloch, Joshua - Effective Java - Pearson Education Limited (US titles) - Addison Wesley Professional (2018)
  - Alexander Shvets - Design Patterns Explained Simply (2013)
  - Alexander Shvets - Dive Into Design Patterns (2019)

Thank you!