Object-Oriented Programming

Syllabus

- Teaching team:
  - Lecturer: Quản Thái Hà
  - Lab instructors: Quản Thái Hà, Phạm Thị Đức
- 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

- Tutorial participation + homework: 20%
- Midterm Examination: 20%
- Final Examination: 60%

Schedule

Week	Lectures	Details
1	Java Programming	- Introduction - 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
2	OOP: OOP Introduction	- From Functions to Objects
3	OOP: Using and designing Objects	- Classes as custom types, objects vs classes, class definition, constructors, access modifiers, this keyword, static data and methods, overloading, modularity, encapsulation/data hiding, immutability - Identifying classes, UML class diagrams
4	Java Programming: 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 - Array, String, StringBuilder, StringBuffer
5	OOP: Inheritance and Polymorphism	- Inheritance (is-a), aggregation (has-a), casting - Overloading, overriding - Super keyword, final keyword - Runtime polymorphism, dynamic binding, instanceof operator

Week	Lectures	Details
6	OOP: Abstraction and ADT	- Abstract method and abstract class, interface, abstract vs interface - ADT (Abstract Data Type)
7	Java Programming	- Java Collections: Java Collections Interfaces, Sets, Queues, Lists, Maps, Iterators, Algorithms
8	Principles of OO Design	- SOLID
9	Design Patterns	- Creational Design Patterns
10	Design Patterns	- Structural Design Patterns
11	Design Patterns	- Behavioral Design Patterns
12	Correctness and Robustness	- Writing Correct Programs - Exceptions
13	Generic Programming	- Generic Programming: Generic Classes, Type Parameters, Generic Methods, Constraining Type Parameters, Wildcards, Type Erasure
14	Design Language Evolution	- Functional Programming: Lambda functions, functions as values, method references, streams
15	GUI	- Concurrency, Event Handling, JavaFX, Scene Builder

- 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
  - James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley, **The Java Language Specification - Java SE 8 Edition**, (2015).
  - Allen B. Downey, Chris Mayfield, **Think Java**, (2016).
  - Graham Mitchell, **Learn Java the Hard Way**, 2nd Edition, (2016).
  - Bloch, Joshua - **Effective Java** - Addison Wesley Professional (2018)
  - Martin Fowler, **UML Distilled - A Brief Guide To The Standard Object Modeling Language**, (2004).
  - Richard Warburton, **Object-Oriented vs. Functional Programming - Bridging the Divide Between Opposing Paradigms**, (2016).
  - Naftalin, Maurice Wadler, Philip, **Java Generics and Collections**, O'Reilly Media, (2009).
  - Alexander Shvets - **Design Patterns Explained Simply**, (2013)
  - Alexander Shvets - **Dive Into Design Patterns**, (2019)

Thank you!