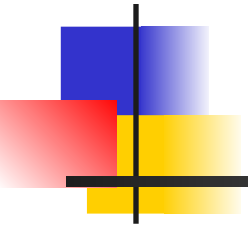


Introduction

Object-Oriented Programming & Modeling with lab
at Kasetsart University

by

James Brucker

Why study object-oriented programming?

- O-O is the dominant programming paradigm
- You'll need it in your internship.
  Many interns say they used OOP knowledge *a lot*.
- Employers require good O-O background.
- Many other courses build on what you learn in OOP.
  - *Without Java, O-O, and UML basics, you will struggle for the next 3 years*.

3 Courses in 1

Enroll 1 Get 2 FREE

Java

Object Design

Modeling with UML

3 Areas We Will Study

these 3 areas are strongly related...

Java	Object Orientation	Modeling
Java syntax and use.	OO approach to program design and construction.	Abstraction.
How to use Java API.	Encapsulation, polymorphism, & inheritance -- how to use them.	Modularity.
Graphical Programs	Design Principles	UML as modeling language.
Collections, generics, & Java object model.	Design Patterns	Modeling of real-world situations using objects.
Frameworks.		

General Goals

Gain understanding and practical skill in...

- O-O paradigm and *why* it matters
- Java programming skill
- software design concepts and design patterns
- Unified Modeling Language (UML) to express design
- software development practices:
  - design before coding
  - unit testing
  - iterative development
- frameworks for developing apps

What You Will Learn

- Java language

```
String [] words = inputline.split( "\\s+" );
List<String> vocab = new ArrayList<String>( );
```

- Programming Guidance

*Source code should be **readable***
Document your code -- explain why, not what

- Idiomatic Usage

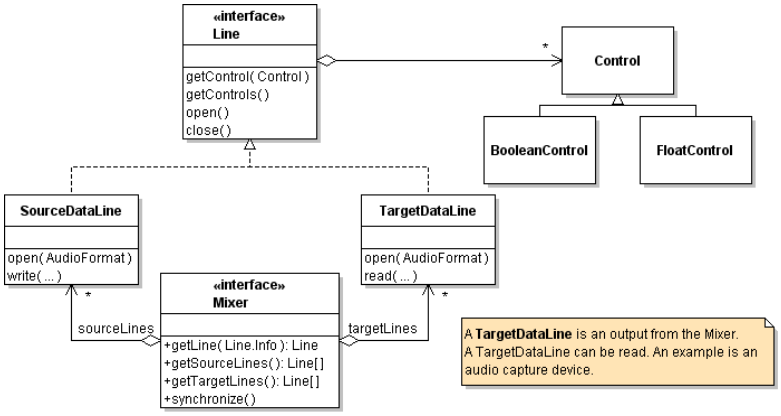
```
for( String word : words )
    if ( ! vocab.contains(word) ) vocab.add( word );
```

- Object Oriented Design Principles

- *Program to an Interface, not an implementation*
- *Don't repeat yourself.*
- *Separate what varies from what stays the same*

What You Will Learn

- Modeling with UML



- Design Patterns

Command	Singleton
Strategy	State
Iterator	Factory Method
Observer	

- Technology & Frameworks

Graphical User Interfaces
Unit testing with JUnit
Charting with JFreeChart

Approach

1. Labs to learn and practice concepts.
2. Mix of topics in labs

   Java language programming style

   OO principles design patterns technology
3. Programming assignments for deeper understanding

Evaluation

One grade for both lecture and lab work.

Your grade is based on:

- Midterm and Final exams
- Laboratory exams (programming)
- Programming assignments
- Class participation
- Quiz scores
- Laboratory participation

At least 50% on <u>both</u> written and lab exams to pass

Approximate Grading Scale

A	85% and above
B	75% - 85%
C	65% - 75%
D	55% - 65%
F	less than 55% overall
	or written exam average < 50%
	or lab exam average < 50%

that means, to pass you must average >= 50% on written exams and lab exams.

Class Homepage and Repository

Schedule and Info

`https://bitbucket.org/skeoop/oop/wiki/Home`

Weekly materials including labs and homework

`https://bitbucket.org/skeoop/oop/src`

or, check them out using Git:

Lab

Schedule: Evening 16:00 - 19:00 Room 201

Please do not bring food into lab.

Alternative: Start at 16:30? (vote)

Why Put in Effort?

We are what we do.

Excellence, therefore, is a habit.

-- Aristotle

Push yourself in this course ...

- *prepare for your career*
- *develop a habit of excellence in anything*
- *maybe get "A"*
- *enjoy your time at KU more*

Why Practice?

I hear and I forget,

I see and I remember,

I do and I understand.

-- Confucious