객체지향 설계 프로젝트
Object Oriented Design Project

Chonnam National University
School of Electronics and Computer Engineering

Kyungbaek Kim

Syllabus

- Instructor
  - Kyungbaek Kim
  - Office : engineering building #6-715
  - Email : kyungbaekkim@jnu.ac.kr
  - Office Hours : Friday 15:00 ~ 16:00
- Lectures : Wed 14:00~15:00, Fri 13:00~15:00
- Location : engineering building #6-105
- Course Web Page : e-Class page on JNU Portal
  - Other page : http://myweb.jnu.ac.kr/~kbkim/jnu_classes/2016f_oo_design_project_index.html

Syllabus (cont'd)

- Textbook
  - Head First Object-Oriented Analysis and Design, Brett D. McLaughlin, Gary Pollice, Dave West
- Reference Book
  - Core Java 2, Volume I Fundamentals (7th Edition), Cay S. Horstmann, Gary Cornel

Grading Policy

- Attendance – 10%
  - 100% attendance will get bonus points
- Quiz and technical evaluation – 25%
  - Tentatively scheduled for 3-4 times of quiz (15%)
  - Technical evaluation of individual students (10%)
- Project – 65 %
  - 4 check points with progress reports (25%)
  - 4 times presentations (40%)
  - Mutual evaluation by students

Project

- The main part of this course is conducting your own project
- 4 students make one team
- Using Java (**v 1.8**), Eclipse, Javadoc, UML, subversion (or Git)and etc.
- You can choose a target program for any application
  - **Mandatory :** You need to implement its GUI with SWING
  - **Mandatory :** You need to provide a runnable JAR file for your application
  - **Mandatory :** Final Program should be correctly operated. Otherwise You will fail this course.
  - **Preferred :** Please use a general file system API to store some information of your program.
  - **Optional :** You can use networking functionality, if you can. But the risk of jeopardizing your application is up to you.
  - e.g. 게임, 금전관리/가계부, 영화/음악/문서/사진관리, 일정관리/일기장, 물류/자재관리, 수업관리 etc.

Example of applications

- GAME
  - 2012f : 2 – Joker RUN
  - 2014f : 2 – 전대마블
  - 2014f : 5 – Balloon Cup
  - 2014f : 7 – SpaceWar 2014
  - 2015f : 1 – Life of Salmon
  - 2015f : 5 – Castle Defence
  - 2015f : 7 – 휴먼그림체
- Utility
  - 2014f : 3 – Music Player
  - 2015f : 2 – File Avenger
  - 2015f : 8 – Students' Guide To Classroom
  - 2015f : 12 – Mapmo
- Data Manager
  - 2012f : 3 – 학점계산기 (pw:5555)
  - 2013f : 3 – 학점 박차오름
  - 2014f : 8 – Product Manager
  - 2015f : 4 – 전컴 도서관

Evaluation of Project

- 4 times checkpoints
  - Initial setup : around 21th September
    - Name, Motivation, Characteristics, Functions
  - Initial design : around 12th October
    - Requirement Lists, Usecases, Usecase Diagram, Sequence Diagram, User Interface Designs, Functional Specification, Work Distribution
  - Beta release : around 16th November
    - Class Diagram, API manuals (with Javadoc), Beta Demo, Progress of projects
    - Other teams provide feedback of your beta release.
  - Final release : around 7th December
    - Updates for feedbacks
    - Evaluation of functional specification, Final Demo
    - Finalizing your project

Evaluation of Project (cont')

- For each checkpoint, each team provides a report and a presentation
  - Report : 5%, 5%, 5%, 10%
  - Presentation : 5%, 10%, 10%, 15%
    - Pick **two times** for English presentation.
      - First and third checkpoints are preferred.
    - Mutual evaluation by students (60%) + Evaluation by Professor (40%)

Quiz

- Tentatively scheduled for 3-4 times of quiz
  - JAVA basic
  - SWING basic
  - Object Oriented Design Concepts

Technical evaluation of individuals

- Evaluating designing and coding ability of personals
  - By personal meeting
  - Coding skill
  - Understanding of software design
  - Workload for team project
  - Tentatively scheduled on the final week

After this class, you will …

- Design and Implement medium scale software with the concept of object-oriented designs.
- Be familiar with the tools of managing projects for Java software such as Eclipse and Subversion.
- Exposed to the experience of using Java SWING and Applets

Lecture Schedule

- Reminder of Java Language
  - Package, Class, Methods and Parameters
  - Collections and Exceptions
- Basic Object Oriented Principles
- Gathering Software Requirements
  - Use cases
- Handling Requirement changes
  - Scenarios
- Class Diagram and UML
  - Loosely coupled applications

Lecture Schedule (cont')

- Solving Big problems
  - Divide and conquer
- Architecture of Software
  - Reducing risk
- Iterating and testing
- Design Principles
  - Open-closed, Don't Repeat Yourself, Single Responsibility, Liskov Substitution

Lecture Schedule (cont')

- Programs keep changing
  - Flexibility
- Minimize the risk of modification
- Good Design = Flexible software

Miscellaneous

- Introducing Eclipse
- Introducing SWING
- Introducing UMLet
- Introducing JavaDoc
- Introducing Subversion, git