COMP 3111
SOFTWARE ENGINEERING

COURSE INFORMATION

COURSE INFORMATION

Instructor

L1	Charles ZHANG	Room 3516	charlesz@ust.hk
L2	Kenneth LEUNG	Room 3548	kwtleung@ust.hk

Teaching Assistants

See course web page.

Course Schedule

COMP3111/3111H L1:	MoWe 09:00AM - 10:20AM	Room 2306
COMP3111/3111H L2:	TuTh 12:00PM - 01:20PM	Room 2502
COMP3111 LA1:	Fr 09:30AM - 11:20AM	Room 4210
COMP3111 LA2:	Th 06:00PM - 07:50PM	Room 4210
COMP3111 LA3:	We 10:30AM - 12:20PM	Room 4210
COMP3111H LA1:	Mo 04:00PM - 05:50PM	Room 4210
COMP3111H LA2:	Mo 10:30AM - 12:20PM	Room 4210

COURSE TEXTBOOK

Reference Textbook

Object-Oriented Software Engineering: Using UML, Patterns, and Java, 3/E,
B. Bruegge and A.H. Dutoit,
Pearson Education, Inc., 2010.

Development Tools, Documentation

Java, Git / GitHub, draw.io

– Lab Notes, Web resources.

COURSE REQUIREMENTS

Item		Value
Exercises (in-class practice exercises)		10%
After you answer exercise questions for 10 lectures, you will obtain full scores (i.e., 10%) for In-class Exercises		
Labs		10%
Project		30%
Activity 1: System Requirements Specification	2%	
Activity 2: System Implementation & Testing	28%	
Final Exam — Dec exam period		50%

COURSE OVERVIEW AND OBJECTIVES

Focus: A *disciplined* approach to software development.

☞ **This course provides both a theoretical foundation and practical skills in software engineering.**

Overall learning objectives:

1. An understanding of the concepts and practices of software engineering.
2. Practical experience analyzing, designing, implementing and testing a software system and working in a development team.

INTENDED LEARNING OUTCOMES

- Ability to apply appropriate modeling techniques to design software for an application of medium complexity.

- Ability to apply appropriate software engineering techniques to implement an application of medium complexity.

- Ability to function effectively as a member of a software development team: organize, manage and participate in a small software development team and plan and schedule the activities involved in developing an application of medium complexity.

WHY SOFTWARE ENGINEERING?

- Learn how to design and engineer a ***software system*** (not just a program).
- Learn to express design ideas ***formally*** using a modeling language.
- Learn interpersonal and team ***communication*** skills.
- Learn project ***management*** skills.
  - workload management
  - people management
- Learn ***leadership*** skills (CTO versus coder).

It's *fun* and *satisfying* to build *useful* software!

SYLLABUS

Lecture Topic	Lectures	
Introduction	1	
Modeling Software Systems using UML	2	----> a modeling language
Software Development	2	----> different approaches
System Requirements Capture	5	----> engineering activities
Implementation	2	
Testing	3	
System Analysis & Design	4	
Software Quality Assurance	1	----> management activities
Managing Software Development	1	

IMPORTANT NOTES AND POLICIES

Instructional approach	→	Learn by *listening* and *doing*.
Expected work load	→	*Appropriate* (*4 credit course*).
Project due dates	→	Strictly enforced!
Labs	→	Learn to use software tools. Implement and test your system.
Academic conduct	→	Be honest! *Copying/cheating will be severely penalized!*
Classroom etiquette	→	Be polite and considerate! (Talking during lectures is impolite.)

COURSE PROJECT

Project Overview

<u>Implement part of a medium-sized Java project</u>

Semester-long	→ apply theories; have fun building a system	
Team-based	→ Activities 1 and 2 (3 person teams) *(May be in different lecture sections. 3111H class students can only team up with H-class students (L1 or L2).)*	
Tool-based	→ draw.io	} software modeling
	Java	} code development
	Git / GitHub	} code management
Schedule-oriented	→ strict deadlines!	

COURSE PROJECT

Project Problem Statement

You are given the system requirements.

– You need to turn requirements into a working system (i.e., code).

Activity 1: System Requirements Specification

– Capture and represent the system requirements using models.

- Team-based.
- Used to document a design and explore design ideas.
- Used to communicate with the client and other developers.

Activity 2: Final System Implementation & Testing

– Implement and test the system requirements.

- Team-based using SCRUM.
- Learn team communication (scrum meetings; meeting minutes).
- Learn project management (sprints; burndown charts).

Project/Requirements Questions?

– Email kevinw@ust.hk

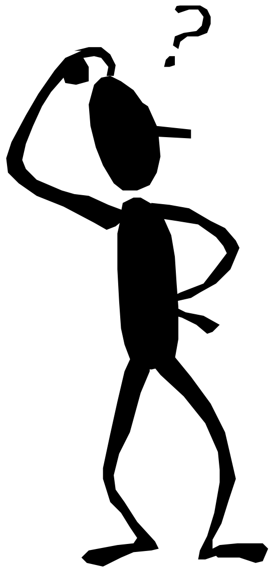
COURSE PROJECT

Project Grading

- Ability to implement requirements correctly according to a schedule.
- Ability to document system requirements using several models.
- For team-based activities individual contribution by team members.
  - ☞ Intra-Peer Evaluation

No freeloading!

WELCOME TO COMP 3111!



Any Questions?