2309 SE251/CS451/CS003 Course Outline

Subject Code	:	**SE251/CS451/CS003**
Subject Title	:	Software Engineering Practices
Course Type	:	Compulsory
Level	:	3
Credits	:	3
Teaching Activity	:	Lab 45 hours
Prior Knowledge*	:	SE240 Data Base System, SE111 Object-Oriented Programming

Class Schedule :

Class	Week	Time	Classroom	Date
D1	WED	12:30-15:20	B401	2023/09/04-2023/12/17

Instructor	:	Tao Zhang
Contact Number	:	(853) 88973009
E-mail Address	:	tazhang@must.edu.mo
Office	:	A321
Office Hour	:	Monday (9:00 - 11:00) Tuesday (9:00 - 11:00) Wednesday (14:30 - 17:30) Thursday (9:00 - 12:00)

Course Description

Through experiments and practice, this course aims to use laboratory sessions to help student understand theories, methods, technologies and general practices for large-scale software development. In practice, how are the basic concepts and theories used?

These concepts include software processes and agile methods, and essential software development activities, from initial specification through to system maintenance. The fundamentals of software engineering will be covered. They include understanding system requirements, finding appropriate engineering compromises, effective methods of design, coding, and testing, team software development, and the application of engineering tools. Project management and professional software engineering practice will also be covered. Case studies provide practical examples for many of these concepts.

- **Textbook(s)**

Book Name: Object-Oriented Software Engineering
Author/Editor: Timothy C. Lethbridge and Robert Laganière
Edition: Second Edition
ISBN: 007124770X
Publisher: McGraw Hill
Date: 2001

References
TK documentation
http://www.tkdocs.com/tutorial/

Ideas for Projects in Tcl/Tk
http://wiki.tcl.tk/14335

Tcl as Unix shell scripts
http://wiki.tcl.tk/4187

Software Engineering Projects
http://www.ece.rutgers.edu/~marsic/books/SE/projects/

INTENDED LEARNING OUTCOMES

Upon successful completion of this subject, students will be able to:

1. investigate, analyze, and use software tools to increase the productivity of software development,
2. understand rapid software development techniques,
3. identify specific components of a software design that can be targeted for reuse,
4. estimate the total cost in a project for software development,
5. conduct software-testing,
6. gain experience in group-based software development and develop communication, planning and time-management skills, and
7. acquire knowledge of professional codes of conduct in software engineering and demonstrate understanding through real-world scenarios.

- **Schedule**

Week	Topic	Hours	Teaching Method
1	A Brief Introduction to Software Engineering	1.0	Lab
	A Brief Introduction to Software Engineering	2.0	Lab
2	User Interface Design	1.0	Lab
	User Interface Design	2.0	Lab
3	Introduction to Prototyping	1.0	Lab
	Introduction to Prototyping	2.0	Lab
4	Requirements and Use-Case Diagrams	1.0	Lab
	Requirements and Use-Case Diagrams	2.0	Lab
5	Cost Estimation	1.0	Lab
	Cost Estimation	2.0	Lab
6	NLP and SE	1.0	Lab
	NLP and SE	2.0	Lab
7	OO modelling: Class modelling	1.0	Lab
	OO modelling: Class modelling	2.0	Lab
8	Tcl/TK as a tool for prototyping	1.0	Lab
	Tcl/TK as a tool for prototyping	2.0	Lab
9	Prototyping in Practice	1.0	Lab
	Prototyping in Practice	2.0	Lab
10	Prototyping for the Project	1.0	Lab
	Prototyping for the Project	2.0	Lab
11	Software Design in Practice	1.0	Lab
	Software Design in Practice	2.0	Lab
12	Case study: Financial application with R	1.0	Lab
	Case study: Financial application with R	2.0	Lab
13	CASE tools	1.0	Lab
	CASE tools	2.0	Lab
14	Finishing the Project	1.0	Lab
	Finishing the Project	2.0	Lab
15	Student Demos and Presentations	1.0	Lab
	Student Demos and Presentations	2.0	Lab

- **ASSESSMENT APPROACH**

Assessment method	% weight
1. Attendance (Class participation)	**10%**
2. Demo Presentation	**30%**
3. Project Report	**60%**
Total	**100 %**

Guideline for Letter Grade:

Marks	Grade
93-100	A+
88-92	A
83-87	A-
78-82	B+
72-77	B
68-71	B-
63-67	C+
58-62	C
53-57	C-
50-52	D
0-49	F
Marks	Grade

Notes:
Students will be assessed on several assessment items (i.e. attendance, assignments, and project work.).
The attendance evaluates the student's participation of discussion in the classes.
The project is used to evaluate the student's ability to apply the knowledge to solve practical problems in software engineering. Students should show their project demo and write the project report as each group.