南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

课程详述

COURSE SPECIFICATION

以下课程信息可能根据实际授课需要或在课程检讨之后产生变动。如对课程有任何疑问，请联系授课教师。

The course information as follows may be subject to change, either during the session because of unforeseen circumstances, or following review of the course at the end of the session. Queries about the course should be directed to the course instructor.

1.	**课程名称 Course Title**	
2.	**授课院系** **Originating Department**	
3.	**课程编号** **Course Code**	
4.	**课程学分 Credit Value**	
5.	**课程类别** **Course Type**	通识必修课程 General Education (GE)Required Courses 通识选修课程 General Education (GE) Elective Courses 专业基础课 Major Foundational Courses 专业核心课 Major Core Courses 专业选修课 Major Elective Courses （请保留相应选项 **Please only keep the relevant information**）
6.	**授课学期** **Semester**	春季 Spring / 夏季 Summer / 秋季 Fall
7.	**授课语言** **Teaching Language**	英文 English
8.	**授课教师、所属学系、联系方式（如属团队授课，请列明其他授课教师）** **Instructor(s), Affiliation& Contact** **（For team teaching, please list all instructors）**	
9.	**实验员/助教、所属学系、联系方式** **Tutor/TA(s), Contact**	无 NA / 待公布 To be announced / 已确定的实验员/助教联系方式 Please list all Tutor/TA(s) （请保留相应选项 **Please only keep the relevant information**）
10.	**选课人数限额(可不填)** **Maximum Enrolment**	

（Optional）	

11. 授课方式 Delivery Method	讲授 Lectures	习题/辅导/讨论 Tutorials	实验/实习 Lab/Practical	其它(请具体注明) Other（Please specify）	总学时 Total
学时数 Credit Hours					

12. 先修课程、其它学习要求 Pre-requisites or Other Academic Requirements	CS309 面向对象分析与设计 Object-oriented Analysis and Design OR an equivalent course in another university
13. 后续课程、其它学习规划 Courses for which this course is a pre-requisite	CS 409 软件测试 Software Testing
14. 其它要求修读本课程的学系 Cross-listing Dept.	无

教学大纲及教学日历 SYLLABUS

15. 教学目标 **Course Objectives**

This course introduces some basic concepts in Software Engineering. Topics covered include unit testing, software configuration management, software process and reverse engineering. Some advanced topics like test generation and static analysis will also be covered. This course emphasizes on software project management, team collaboration and usage of version control system. Students will apply the knowledge they learned in various lab individual assignments. Students will also apply their knowledge on project management and collaboration through version control system in a course project that last for 8 weeks.

16. 预达学习成果 **Learning Outcomes**

Upon completion of this course, the students are expected to have a good understanding of writing unit tests, using software configuration management system, applying software process and reverse engineering in real-world applications.

17. 课程内容及教学日历 （如授课语言以英文为主，则课程内容介绍可以用英文；如团队教学或模块教学，教学日历须注明主讲人）

Course Contents (in Parts/Chapters/Sections/Weeks. Please notify name of instructor for course section(s), if this is a team teaching or module course.)

Week 1: Introduction

- Introduction to course
- Introduction to Software Engineering

[Lab] Introduction to different roles in software projects, including project manager, software engineering and software tester.

Week 2: Software Configuration Management

- Introduction to concepts in software configuration management
- Example of coftware configuration management systems (CVS, SVN, Git)
- Explain the differences between different systems and how to use these systems

[Lab] Learn about the commands in Git and the basics of using GitHub for managing a software project.

Week 3: Extreme Programming

- History of XP
- Introduce some general practices of XP (Planning Game, pair programming and user stories)
- Provide examples and benefits of planning poker and pair programming.

[Lab] Learn about how to use Github classroom for submitting programming assignments and the syntax of markdown in GitHub,

Week 4: Concepts in Extreme Programming and Introduction to Testing

- Explain about the role of testing in XP
- Introduce the motivations behind testing
- Provide examples and benefits of testing.

[Lab] Learn about how to write JUnit test and practice the concept of “pair testing”

Week 5: Test-driven Development

- Introduce the steps in Test-driven development
- Provide examples and benefits of test-driven development
- Introduce the concept of code coverage and explain how to compute different coverage metrics

[Lab] Learn about how to use learn about how to use automated test generation tool for generating tests and how to measure test coverage using Jacoco plugin

Week 6: Mutation testing and Test generation

- Introduce the concepts of mutation testing
- Introduce the concepts of test generation
- Introduce the test generation algorithm of Evosuite.

[Lab] Learn about the basics of tensorflow API and build a simple mobile app that uses Tensorflow-Lite.

Week 7: Software metrics

- Introduce the concepts of software metrics
- Introduce technical and non-technical software metrics
- Explain coupling and cohesion

[Lab] Learn about the basics of tensorflow API and build a simple mobile app that uses Tensorflow-Lite.

Week 8: Reverse Engineering

- Introduce the concepts of mutation testing.
- Introduce some re-engineering patterns.
- Explain some reverse engineering activities.

[Lab] Learn about the basic of static analysis tools and integrate these tools to the course project, .

Week 9: Static Analysis

- Introduce the concepts of static analysis.
- Introduce coding standard and common bug patterns.
- Explain the benefits of static analysis compared to dynamic analysis.

[Lab] Learn about how to reverse engineer an Android app using apktool and Java Decompiler.

Week 10: Component and Reuse

- Introduce the concepts of component and framework.
- Explain the benefits of software reuse
- Explain the techniques in improve software reuse

[Lab] Learn about some popular frameworks and examples of software reuse. Group discussion on course project.

Week 11: UI design

- Introduce the concepts of good UI designs.
- Explain good UI design through several examples.
- Explain how the UI design influence the design of code and tests.

[Lab] Compare some UI designs and choose a good UI design for the course project

Week 12: Continuous Integration & Regression Testing

- Introduce the concepts of continuous integration.
- Introduce the concepts of regression testing.
- Explain the tool supports and techniques for continuous integration and regression testing.

[Lab] Learn about how to use configure software projects for continuous integration server and write good regression test cases.

Week 13: Documentation

- Introduce the concepts of good software documentation.
- Explain formal and informal documentation.
- Introduce research on software documentation

[Lab] Learn about how to write good Javadoc comments and prepare user manual and documentation for course project.

Week 14: Review for Final exams

- Review all concepts thought in the classes

- Provide some class exercises as a preparation for final exam
- Provide guidelines on course project report and presentation.

[Lab] Group discussion on course project and prepare report for the course project.

Week 15: Course Project Presentation 1

- Students will give a presentation on course project.
- Discuss the pros and cons of each project.
- Provide feedback for improving the writing and future presentation skills.

[Lab] Prepare the source code and user manuals for the final release.

Week 15: Course Project Presentation 2

- Students from different group will give a presentation on course project.
- Discuss the pros and cons of each project.
- Provide feedback for improving the writing and future presentation skills.

[Lab] Prepare the final report and future works for the course project.

18. 教材及其它参考资料 **Textbook and Supplementary Readings**

Code Complete: A Practical Handbook of Software Construction (Second Edition) by Steve McConnell.

Software Engineering (10th Edition) by Ian Sommerville.

Object-Oriented Reengineering Patterns by Oscar Nierstrasz, Stéphane Ducasse, Serge Demeyer

Various articles in journals and conference proceedings given during the lectures.

课程评估 ASSESSMENT

19. 评估形式 **Type of Assessment**	评估时间 **Time**	占考试总成绩百分比 **% of final score**	违纪处罚 **Penalty**	备注 **Notes**
出勤 **Attendance**		10%		In-Class Exercises/Attendance
课堂表现 **Class Performance**		20%		Weekly lab tutorial
小测验 **Quiz**				
课程项目 **Projects**		35%		
平时作业 **Assignments**				
期中考试 **Mid-Term Test**				
期末考试 **Final Exam**		35%		

期末报告 Final Presentation				
其它（可根据需要改写以上评估方式） Others (The above may be modified as necessary)				

20. 记分方式 **GRADING SYSTEM**

☑ A. 十三级等级制 **Letter Grading**
☐ B. 二级记分制（通过/不通过） **Pass/Fail Grading**

课程审批 REVIEW AND APPROVAL

21. 本课程设置已经过以下责任人/委员会审议通过
This Course has been approved by the following person or committee of authority