City University of Hong Kong
Course Syllabus

offered by Department of Computer Science
with effect from Semester A 2024/25

Part I   **Course Overview**

Course Title:	Software Engineering
Course Code:	CS5351
Course Duration:	One semester
Credit Units:	3 credits
Level:	P5
Medium of Instruction:	English
Medium of Assessment:	English
Prerequisites: *(Course Code and Title)*	CS2312 Problem Solving and Programming or equivalent
Precursors: *(Course Code and Title)*	Nil
Equivalent Courses: *(Course Code and Title)*	Nil
Exclusive Courses: *(Course Code and Title)*	Nil

Part II   Course Details

1.   Abstract

The aim of this graduate-level course is to provide students with a comprehensive understanding of the state-of-art and practices in the software engineering (SE) discipline, its associated processes/methodologies and current trends.   This includes in-depth coverage of some of the key SE issues, best practices and guidelines and an overview of project management techniques.   The key objective is to equip students with SE knowledge so that they will be able to take full advantage of these concepts, processes, and best practices in their future software development projects.

2.   Course Intended Learning Outcomes (CILOs)
(CILOs state what the student is expected to be able to do at the end of the course according to a given standard of performance.)

No.	CILOs	Weighting (if applicable)	Discovery-enriched curriculum related learning outcomes (please tick where appropriate)		
			A1	*A2*	*A3*
1.	Describe fundamental software engineering process models and comprehend the current trends		✓		
2.	Explain and analyse advanced software engineering principles and techniques			✓	
3.	Create and maintain the design and codebase of software engineering projects			✓	
4.	Perform team-based software engineering tasks.		✓		
		100%			

A1:   Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.
A2:   Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to real-life problems.
A3:   *Accomplishments*
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

3. Learning and Teaching Activities (LTAs)
(LTAs designed to facilitate students' achievement of the CILOs.)

LTA	Brief Description	CILO No. 1	2	3	4	Hours/week (if applicable)
Lectures	Students will engage with selected software engineering methodologies and advanced techniques and principles. These elements will be illustrated with principles, examples, and demonstration.	✓	✓	✓		2 hours/ week
Tutorials	Students will discuss and practice various principles and skills in software developments and project management in a controlled software engineering context.	✓	✓	✓		1 hour/ week
Reading	Students will read from assigned software engineering materials and explain the technicality with original ideas for advancing the state of the art.	✓	✓			
Group Project	Students as a team will design and create a system for software engineering projects. They will collaborate and share in their learning process. They will also practice materials in all major topics from project management and requirements to coding and testing in the project.	✓	✓	✓	✓	

4. Assessment Tasks/Activities (ATs)
(ATs are designed to assess how well the students achieve the CILOs.)

Assessment Tasks/Activities	CILO No. 1	2	3	4	Weighting	Remarks
Continuous Assessment: 60%						
Quiz	✓	✓	✓		20%	
Research Paper Presentation		✓			10%	
Project	✓	✓	✓	✓	30%	
Examination^: 40% (duration: 2 hours)						
					100%	

^ For a student to pass the course, at least 30% of the maximum mark for the examination must be obtained.

5. Assessment Rubrics

(Grading of student achievements is based on student performance in assessment tasks/activities with the following rubrics.)

Applicable to students admitted before Semester A 2022/23 and in Semester A 2024/25 & thereafter

Assessment Task	Criterion	Excellent (A+, A, A-)	Good (B+, B, B-)	Fair (C+, C, C-)	Marginal (D)	Failure (F)
1. Quiz	1.1 ABILITY to describe, analyse and apply software engineering processes and techniques.	High	Significant	Moderate	Basic	Not even reaching marginal levels
2. Research Paper Presentation	2.1 ABILITY to explain and present technical contents in software engineering research papers.	High	Significant	Moderate	Basic	Not even reaching marginal levels
3. Project	3.1 ABILITY to apply software engineering techniques and good practices for actual software development. 3.2 ABILITY to use project management skills in team project. 3.3 ABILITY to report in an organised and logical way. All works are professional presented. All sources are correctly and thoroughly documented.	High	Significant	Moderate	Basic	Not even reaching marginal levels
4. Examination	4.1 ABILITY to explain software development processes and compare process models. 4.2 ABILITY to apply software analysis and design techniques. 4.3 ABILITY to apply project management techniques on given case study. 4.4 ABILITY to apply good software engineering practices on given case study.	High	Significant	Moderate	Basic	Not even reaching marginal levels

Applicable to students admitted from Semester A 2022/23 to Summer Term 2024

Assessment Task	Criterion	Excellent (A+, A, A-)	Good (B+, B)	Marginal (B-, C+, C)	Failure (F)
1. Quiz	1.1 ABILITY to describe, analyse and apply software engineering processes and techniques.	High	Significant	Basic	Not even reaching marginal levels
2. Research Paper Presentation	2.1 ABILITY to explain and present technical contents in software engineering research papers.	High	Significant	Basic	Not even reaching marginal levels
3. Project	3.1 ABILITY to apply software engineering techniques and good practices for actual software development. 3.2 ABILITY to use project management skills in team project. 3.3 ABILITY to report in an organised and logical way. All works are professional presented. All sources are correctly and thoroughly documented.	High	Significant	Basic	Not even reaching marginal levels
4. Examination	4.1 ABILITY to explain software development processes and compare process models. 4.2 ABILITY to apply software analysis and design techniques. 4.3 ABILITY to apply project management techniques on given case study. 4.4 ABILITY to apply good software engineering practices on given case study.	High	Significant	Basic	Not even reaching marginal levels

Part III   Other Information (more details can be provided separately in the teaching plan)

1. **Keyword Syllabus**
    (An indication of the key topics of the course.)

    Overview of the software engineering discipline. Software engineering process models and trends. Software engineering standards, techniques, best practices, and guidelines. Software project management.

    Syllabus
    1. Overview of the Software Engineering Discipline
        History and overview of the software engineering discipline. Major roles, issues and problems. Current trends and directions.

    2. Software Engineering Processes and Techniques
        Overview of different software engineering process models, such as structured analysis and design, object-oriented analysis and design, agile methodologies, coding, maintenance, and testing, and their trends. Contrasting and comparing the different models and techniques.   The individual processes within the process models (such as requirements, implementation, testing, etc.), their roles, issues, deliverables (both diagrams, documents and software), quality management and project management.

    3. Software Engineering Standards, Best Practices, and Guidelines
        Overview of different SE-related standards, best practices, and guidelines, such as those provided by IEEE, ACM, SEI, etc.

    4. Software Project Management
        Overview of project management concepts as they relate to software engineering, such as those outlined by PMI (and its PMIBOK), for example scope, schedule development, costing and quality management.

2. **Reading List**
2.1 **Compulsory Readings**
    (Compulsory readings can include books, book chapters, or journal/magazine articles. There are also collections of e-books, e-journals available from the CityU Library.)

    There is no textbook for this course. Classic and latest software engineering research papers will be discussed in the lectures and tutorial sessions.

2.2 **Additional Readings**
    (Additional references for students to learn to expand their knowledge about the subject.)

1.	*Software Engineering Institute: http://www.sei.cmu.edu/*
2.	*Project Management Institute: http://www.pmi.org/*
3.	*IEEE SE Online: http://www.computer.org/portal/site/seportal/*
4.	*IEEE SE Standards: http://standards.ieee.org/software/*
5.	*IEEE/ACM SE Code of Ethics and Professional Practice: http://www.acm.org/about/se-code*