The Hong Kong University of Science and Technology

UG Course Syllabus (Spring 2025-26)

[Course Title] Software Engineering

[Course Code] COMP 3111

[No. of Credits] 4

[Any pre-/co-requisites] Prerequisite(s): COMP 2012 OR COMP 2012H. Exclusion(s): COMP 3111H, ISOM 3210

Name: Charles ZHANG

Email: charlesz@cse.ust.hk

Course Description

Methods and tools for planning, designing, implementing, validating, and maintaining large software systems. Project work to build a software system as a team, using appropriate software engineering tools and techniques.

Mode of Delivery: [BLD] Blended learning

Intended Learning Outcomes (ILOs)

By the end of this course, students should be able to:

1. Apply appropriate modeling techniques to design the software for an application of medium complexity: specify, model and document the data and processing requirements for an application using a modeling language and implement a disciplined software design for an application.
2. Apply appropriate software engineering techniques to implement an application of medium complexity: transform requirements for an application, expressed in a modeling language, into a working software system, select appropriate algorithms to implement the stated requirements and design and implement a variety of tests to validate that a software system meets its stated requirements.
3. 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 software for a medium-sized application.
4. Manage a software development environment by selecting appropriate software components to implement the stated requirements for a software system and configuring and managing a software development environment.

Assessment and Grading

This course will be assessed using criterion-referencing and grades will not be assigned using a curve. Detailed rubrics for each assignment are provided below, outlining the criteria used for evaluation.

Assessment Task	Contribution to Overall Course grade (%)
Pre-Lecture Quizzes	5%
In-Class Exercises	5%
Labs	10%
Midterm	20%
Group Project	20%
Final Examination	40%

Assessments:

[List specific assessed tasks, exams, quizzes, their weightage, and due dates; perhaps, add a summary table as below, to precede the details for each assessment.]

Assessment Task	Contribution to Overall Course grade (%)	Due date
Pre-Lecture Quizzes	5%	Before each lecture*
In-Class Exercises	5%	Immediately after each class*
Labs	10%	Immediately after each lab*
Midterm	20%	Week8-Week10*
Group Project	20%	By the end of the semester*
Final Examination	40%	Final Exam Period*

* Assessment marks for individual assessed tasks will be released within two weeks of the due date.

Mapping of Course ILOs to Assessment Tasks

Assessed Task	Mapped ILOs	Explanation
Pre-Lecture Quizzes	ILO4	

		ILO4: Students are assessed on their understanding of appropriate software components, version control, continuous integration, and other foundational concepts in managing a development environment.
In-Class Exercises	ILO4	ILO4: Reinforces practical skills for configuring and managing real-world software development environments in tools like GitHub, Docker, and IDEs.
Labs	ILO4	ILO4: Students gain experience in setting up, configuring, and managing the software environments necessary for different stages of software development (e.g., testing frameworks, CI pipelines).
Midterm	ILO4	ILO4: Tests students' ability to analyze, select, and explain the configuration of development tools and software components in a given scenario.
Group Project	ILO1, ILO2, ILO3, ILO4	ILO1: Students work together to analyze requirements, create system models (e.g., UML), and document their software design using formal modeling techniques. ILO2: Students implement the designed system by transforming models into code, selecting and applying appropriate algorithms, and developing comprehensive test suites. ILO3: Students plan and execute the project collaboratively, demonstrating effective teamwork, task scheduling, and use of collaborative tools like version control. ILO4: While not the primary focus, students will also set up and manage the development environment necessary for team-based project execution.
Final Examination	ILO4	ILO4: Evaluates students' understanding of best practices for

		selecting, configuring, and maintaining development environments and tools throughout the software lifecycle.

Grading Rubrics

Assessment Task	Excellent (A)	Good (B)	Satisfactory (C)	Marginal/Fail (D/F)
Pre-Lecture Quizzes (5%)	Consistently demonstrates strong understanding of tools and environments; answers are accurate and insightful.	Generally correct responses with good grasp of development tools.	Basic understanding shown; minor conceptual errors.	Incorrect or incomplete responses; lacks understanding of tools.
In-Class Exercises (5%)	Completes all tasks accurately; actively applies development tools and configurations.	Most tasks completed correctly; shows solid understanding of environment management.	Some correct tasks; understanding of tools is basic.	Major tasks missing or incorrect; weak grasp of development tools.
Labs (10%)	Successfully configures and manages complex development environments with precision.	Correct setup and management with minor guidance.	Basic configurations completed; some errors present.	Unable to complete setups or manage tools effectively.
Midterm (20%)	Demonstrates deep understanding of selecting and managing appropriate tools; applies concepts clearly.	Good overall understanding with minor mistakes.	Basic knowledge evident but lacks clarity or precision.	Poor grasp of concepts; many incorrect or missing answers.
Group Project (20%)	Excellent system modeling, implementation, and team collaboration; full testing and documentation provided.	Good design and implementation; works well in team; most components tested and documented.	Functional output with acceptable team effort; some modeling or testing incomplete.	Poor teamwork or weak implementation; lacks modeling, testing, or documentation.
Final Examination (40%)	Mastery in tool selection, configuration, and lifecycle management; applies knowledge across scenarios.	Good understanding with minor omissions or misapplications.	Meets minimum expectations with superficial understanding.	Fails to demonstrate competency in software development environment management.

Final Grade Descriptors:

[As appropriate to the course and aligned with university standards]

Grades	**Short Description**	**Elaboration on subject grading description**
A	Excellent Performance	[Example: Demonstrates a comprehensive grasp of subject matter, expertise in problem-solving, and significant creativity in thinking. Exhibits a high capacity for scholarship and collaboration, going beyond core requirements to achieve learning goals.]
B	Good Performance	[Example: Shows good knowledge and understanding of the main subject matter, competence in problem-solving, and the ability to analyze and evaluate issues. Displays high motivation to learn and the ability to work effectively with others.]
C	Satisfactory Performance	[Example: Possesses adequate knowledge of core subject matter, competence in dealing with familiar problems, and some capacity for analysis and critical thinking. Shows persistence and effort to achieve broadly defined learning goals.]
D	Marginal Pass	[Example: Has threshold knowledge of core subject matter, potential to achieve key professional skills, and the ability to make basic judgments. Benefits from the course and has the potential to develop in the discipline.]
F	Fail	[Example: Demonstrates insufficient understanding of the subject matter and lacks the necessary problem-solving skills. Shows limited ability to think critically or analytically and exhibits minimal effort towards achieving learning goals. Does not meet the threshold requirements for professional practice or development in the discipline.]

Course AI Policy

Generative AI is allowed for the course project only.

Communication and Feedback

Assessment marks for individual assessed tasks will be communicated via Canvas within two weeks of submission. Feedback on assignments will include [specific details, e.g., strengths, areas for improvement]. Students who have further questions about the feedback including marks should consult the instructor within five working days after the feedback is received.

Resubmission Policy

All students are required to adhere strictly to the deadlines posted on Canvas. Late submissions will not be accepted under any circumstances, unless prior approval has been granted for exceptional cases. Please plan your time carefully and check Canvas regularly for assignment due dates.

Required Texts and Materials

N/A

Academic Integrity

Students are expected to adhere to the university's academic integrity policy. Students are expected to uphold HKUST's Academic Honor Code and to maintain the highest standards of academic integrity. The University has zero tolerance of academic misconduct. Please refer to Academic Integrity | HKUST – Academic Registry for the University's definition of plagiarism and ways to avoid cheating and plagiarism.

Additional Resources

N/A