

**City University of Hong Kong  
Course Syllabus**

**offered by Department of Computer Science  
with effect from Semester B 2020/21**

---

---

**Part I Course Overview**

**Course Title:** Software Testing and Maintenance

**Course Code:** CS3346

**Course Duration:** 1 semester

**Credit Units:** 3

**Level:** B3

**Proposed Area:**  Arts and Humanities  
(for GE courses only)  Study of Societies, Social and Business Organisations  
 Science and Technology

**Medium of Instruction:** English

**Medium of Assessment:** English

**Prerequisites:** CS3342 Software Design  
(Course Code and Title) or CS3343 Software Engineering Practice  
or equivalent

**Precursors:** Nil  
(Course Code and Title)

**Equivalent Courses:** Nil  
(Course Code and Title)

**Exclusive Courses:** Nil  
(Course Code and Title)

---

## Part II Course Details

### 1. Abstract

(A 150-word description about the course)

This course aims to equip students with the theories, principles and practices of software testing and maintenance. Students have the opportunity of getting familiar with formal approaches, practical techniques and contemporary tools for designing testing strategies and systematically generating test cases for software applications.

### 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 <sup>#</sup>	Weighting* (if applicable)	Discovery-enriched curriculum related learning outcomes (please tick where appropriate)		
			A1	A2	A3
1.	Describe the roles and processes of testing and maintenance within a software life cycle.		✓		
2.	Recognize and discuss the basic issues and fundamental principles in software testing and maintenance.		✓		
3.	Design and implement strategies for testing software in structured and organized ways.			✓	
4.	Apply appropriate techniques and tools to generate test cases systematically for software applications.			✓	

\* If weighting is assigned to CILOs, they should add up to 100%.

100%

<sup>#</sup> Please specify the alignment of CILOs to the Gateway Education Programme Intended Learning outcomes (PILOs) in Section A of Annex.

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 self-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. Teaching and Learning Activities (TLAs)

(TLAs designed to facilitate students' achievement of the CILOs.)

Teaching pattern:

Suggested lecture/tutorial/laboratory mix: 2 hours lecture; 1 hour tutorial.

TLA	Brief Description	CILO No.				Hours/week (if applicable)
		1	2	3	4	
Lecture	Explain key concepts, theories, techniques and practices. Discuss basic issues and fundamental principles.	✓	✓	✓		
Tutorial	Discuss key concepts and issues via short questions. Practice with simple problems and discuss solutions.		✓	✓	✓	
Practical testing exercise	Require students to design and implement strategies for structured and organized testing of a practical software application, apply appropriate techniques and tools to generate test cases systematically, and possibly create customized tools to automate part of the testing tasks. May also require students to write programs for part of a software application and then test the code written by themselves or by others.			✓	✓	

### 4. Assessment Tasks/Activities (ATs)

(ATs are designed to assess how well the students achieve the CILOs.)

Assessment Tasks/Activities	CILO No.				Weighting*	Remarks
	1	2	3	4		
Continuous Assessment: <u>40%</u>						
Assignments			✓	✓	25%	Students are required to work on assignments or exercises at least once every four weeks
Exercise or quiz		✓	✓	✓	15%	
Examination <sup>^</sup> : <u>60%</u> (duration: 2 hours)	✓	✓	✓	✓	60%	
* The weightings should add up to 100%.					100%	

<sup>^</sup> 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.)*

Assessment Task	Criterion	Excellent (A+, A, A-)	Good (B+, B, B-)	Fair (C+, C, C-)	Marginal (D)	Failure (F)
1. Assignment	1.1 ABILITY to DESIGN and IMPLEMENT appropriate strategies for testing software in structured and organized ways 1.2 ABILITY to APPLY appropriate techniques and tools to GENERATE test cases systematically for software applications	High	Significant	Moderate	Basic	Not even reaching marginal levels
2. Exercise or quiz	2.1 ABILITY to ACHIEVE the respective CILOs	High	Significant	Moderate	Basic	Not even reaching marginal levels
3. Examination	3.1 ABILITY to ACHIEVE the respective CILOs	High	Significant	Moderate	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.)

Software testing process. Test case selection. Testing principles, approaches, strategies and objectives. Specification-based testing. Boundary value analysis. Equivalence partitioning. Category-partition testing. Combinatorial testing. Statistical and random testing. Operational profiles. Code-based testing. Control flow analysis. Code coverage. Predicate testing. Static and dynamic data flow analysis. Anomaly detection. Software maintenance process. Software evolution. Regression testing. Maintenance of test suites.

#### 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.)

1.	G.J. Myers, T. Badgett and C. Sandler (2012). <i>The Art of Software Testing</i> . Wiley, 3rd edition.
2.	L. Copeland (2004). <i>A Practitioner's Guide to Software Test Design</i> . Artech House.
3.	Selected articles from <i>IEEE</i> and <i>ACM</i> periodicals: accessible online via CityU library.

##### 2.2 Additional Readings

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

1.	A.P. Mathur (2008). <i>Foundations of Software Testing</i> . Pearson Education, 1st edition.
2.	P. Grubb and A.A. Takang (2003). <i>Software Maintenance: Concepts and Practice</i> . World Scientific, 2nd edition.
3.	R.V. Binders (1999). <i>Testing Object-Oriented Systems: Models, Patterns, and Tools</i> . Addison-Wesley.
4.	Selected documents from <i>IEEE Standards</i> : accessible online via CityU library.