

CS3346: SOFTWARE TESTING AND MAINTENANCE

Effective Term

Semester A 2022/23

Part I Course Overview

Course Title

Software Testing and Maintenance

Subject Code

CS - Computer Science

Course Number

3346

Academic Unit

Computer Science (CS)

College/School

College of Engineering (EG)

Course Duration

One Semester

Credit Units

3

Level

B1, B2, B3, B4 - Bachelor's Degree

Medium of Instruction

English

Medium of Assessment

English

Prerequisites

CS3342 Software Design
or CS3343 Software Engineering Practice

Precursors

Nil

Equivalent Courses

Nil

Exclusive Courses

Nil

Part II Course Details

Abstract

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.

Course Intended Learning Outcomes (CILOs)

CILOs		Weighting (if DEC-A1 DEC-A2 DEC-A3 app.)			
1	Describe the roles and processes of testing and maintenance within a software life cycle.		x		
2	Recognize and discuss the basic issues and fundamental principles in software testing and maintenance.		x		
3	Design and implement strategies for testing software in structured and organized ways.			x	
4	Apply appropriate techniques and tools to generate test cases systematically for software applications.			x	

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.

Teaching and Learning Activities (TLAs)

TLAs	Brief Description	CILO No.	Hours/week (if applicable)	
1	Lecture	Explain key concepts, theories, techniques and practices. Discuss basic issues and fundamental principles.	1, 2, 3	3 hours per week
2	Tutorial	Discuss key concepts and issues via short questions. Practice with simple problems and discuss solutions.	2, 3, 4	8 hours per semester

3	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.	3, 4	After class
---	----------------------------	---	------	-------------

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks (e.g. Parameter for GenAI use)
1	Assignments	3, 4	25	Students are required to work on assignments or exercises after class at least once every four weeks
2	Exercise or quiz	2, 3, 4	15	

Continuous Assessment (%)

40

Examination (%)

60

Examination Duration (Hours)

2

Additional Information for ATs

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

Assessment Rubrics (AR)**Assessment Task**

Assignment

Criterion

- 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

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Not even reaching marginal levels

Assessment Task

Exercise or quiz

Criterion

2.1 ABILITY to ACHIEVE the respective CILOs

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Not even reaching marginal levels

Assessment Task

Examination

Criterion

3.1 ABILITY to ACHIEVE the respective CILOs

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Not even reaching marginal levels

Part III Other Information**Keyword Syllabus**

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.

Reading List**Compulsory Readings**

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

Additional Readings

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