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)

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

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)

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

After class

Assessment Tasks / Activities (ATs)

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

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
**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

Reading List

Compulsory Readings

<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>3     Selected articles from IEEE and ACM periodicals: accessible online via CityU library.</td>
</tr>
</tbody>
</table>

Additional Readings

<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>4     Selected documents from IEEE Standards: accessible online via CityU library.</td>
</tr>
</tbody>
</table>