

## Subject Description Form

<b>Subject Code</b>	COMP3233						
<b>Subject Title</b>	Software Testing and Quality Assurance						
<b>Credit Value</b>	3						
<b>Level</b>	4						
<b>Pre-requisite / Co-requisite/ Exclusion</b>	Pre-requisite: COMP3211						
<b>Objectives</b>	<ul style="list-style-type: none"> <li>• To present the concepts, techniques and metrics for quality assurance in software development.</li> <li>• To develop a good understanding of issues, techniques and tools for software testing.</li> <li>• To enable students to gain a working knowledge of techniques for management of testing projects.</li> </ul>						
<b>Intended Learning Outcomes</b>	<p>Upon completion of the subject, students will be able to:</p> <p><i>Professional/academic knowledge and skills</i></p> <p>(a) appreciate the importance of software quality assurance;</p> <p>(b) apply software testing techniques for information systems development;</p> <p>(c) know the inputs and deliverables of the testing process;</p> <p><i>Attributes for all-roundedness</i></p> <p>(d) work together as a team;</p> <p>(e) communicate in writing a technical document;</p> <p>(f) communicate effectively in English for general project presentation.</p>						
<b>Subject Synopsis/ Indicative Syllabus</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Topic</th> </tr> </thead> <tbody> <tr> <td>1. Software quality assurance Quality factors; cost of quality.</td> </tr> <tr> <td>2. Testing fundamentals Understanding defects; testing concepts; levels of testing; test process</td> </tr> <tr> <td>3. Code-based techniques Control flow and data flow testing; mutation testing; domain testing; error-oriented testing.</td> </tr> <tr> <td>4. Specification-based techniques Equivalence partitioning; boundary value testing; state machine testing;</td> </tr> </tbody> </table>		Topic	1. Software quality assurance Quality factors; cost of quality.	2. Testing fundamentals Understanding defects; testing concepts; levels of testing; test process	3. Code-based techniques Control flow and data flow testing; mutation testing; domain testing; error-oriented testing.	4. Specification-based techniques Equivalence partitioning; boundary value testing; state machine testing;
Topic							
1. Software quality assurance Quality factors; cost of quality.							
2. Testing fundamentals Understanding defects; testing concepts; levels of testing; test process							
3. Code-based techniques Control flow and data flow testing; mutation testing; domain testing; error-oriented testing.							
4. Specification-based techniques Equivalence partitioning; boundary value testing; state machine testing;							

	<p>program verification.</p> <p>5. System testing techniques Configuration testing; Compatibility testing; Usability testing; Web Testing; Security testing.</p> <p>5. Inspection technique Team and roles; process.</p> <p>6. Test tools Test generation tools; Test automation tools; code coverage tool; defect tracking tools.</p> <p>7. Measuring software quality Product metrics; process metrics; GQM; testing maturity model.</p>																																																				
<p><b>Teaching/Learning Methodology</b></p>	<p>The software testing techniques and quality assurance concepts will be covered in the lectures. In the tutorials, students will work on exercises and case studies on software testing techniques. The tutorial will also cover common software testing tools (e.g. unit testing, coverage measurement, GUI testing, performance testing, security testing).</p>																																																				
<p><b>Assessment Methods in Alignment with Intended Learning Outcomes</b></p>	<table border="1" data-bbox="483 919 1398 1402"> <thead> <tr> <th rowspan="2">Specific assessment methods/tasks</th> <th rowspan="2">% weighting</th> <th colspan="6">Intended subject learning outcomes to be assessed (Please tick as appropriate)</th> </tr> <tr> <th>a</th> <th>b</th> <th>c</th> <th>d</th> <th>e</th> <th>f</th> </tr> </thead> <tbody> <tr> <td>1. Assignments</td> <td rowspan="3">55%</td> <td>✓</td> <td>✓</td> <td>✓</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2. Project</td> <td></td> <td>✓</td> <td></td> <td>✓</td> <td>✓</td> <td>✓</td> </tr> <tr> <td>3. Mid-term</td> <td>✓</td> <td>✓</td> <td>✓</td> <td></td> <td></td> <td></td> </tr> <tr> <td>4. Examination</td> <td>45%</td> <td>✓</td> <td>✓</td> <td>✓</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Total</td> <td>100 %</td> <td colspan="6"></td> </tr> </tbody> </table> <p>Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:</p> <p>Students are required to work as a team on a project related to software testing and software quality assurance. This can be used to assess the students on their (b) understanding in software testing techniques for information systems development. Also, the students will be assessed on their ability to (d) work together as a team in preparing a report, (e) writing technical documents, and (f) communicate effectively in English for general project presentation.</p> <p>Assignment(s), mid-term(s) and the final examination will be used to assess the students on their academic knowledge and skills in software testing, which include the ability to (a) appreciate the importance of software quality assurance, (b) apply software testing techniques for information systems development and (c) knowledge in the inputs and deliverables of the testing process.</p>	Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)						a	b	c	d	e	f	1. Assignments	55%	✓	✓	✓				2. Project		✓		✓	✓	✓	3. Mid-term	✓	✓	✓				4. Examination	45%	✓	✓	✓				Total	100 %						
Specific assessment methods/tasks	% weighting			Intended subject learning outcomes to be assessed (Please tick as appropriate)																																																	
		a	b	c	d	e	f																																														
1. Assignments	55%	✓	✓	✓																																																	
2. Project			✓		✓	✓	✓																																														
3. Mid-term		✓	✓	✓																																																	
4. Examination	45%	✓	✓	✓																																																	
Total	100 %																																																				

<b>Student Study Effort Expected</b>	Class contact:	
	▪ Lecture	39 Hrs.
	▪ Tutorial	0 Hrs.
	Other student study effort:	
	▪ Assignments and project	40 Hrs.
	▪ Review	30 Hrs.
	Total student study effort	109 Hrs.
<b>Reading List and References</b>	Reference Books: 1. Ron Patton, Software Testing (2nd Edition), Sams Publishing , 2005 2. Hung Q. Nguyen, Bob Johnson, Michael Hackett and Robert Johnson , Testing Applications on the Web: Test Planning for Mobile and Internet-Based Systems (Second Edition), John Wiley, 2003 3. Rick D. Craig and Stefan P. Jaskiel, Systematic Software Testing, Artech House Publishers, 2002 4. Nina S. Godbole , Software Quality Assurance: Principles And Practice, Alpha Science International, Ltd, 2004	