Subject Description Form

Subject Code	COMP302
Subject Title	Foundations of Software Engineering
Credit Value	3
Level	3
Pre-requisite / Co-requisite/ Exclusion	Pre-requisite: COMP201, COMP210 Co-requisite/ Exclusion: Nil
Objectives	This subject provides students with: • a general knowledge of the application of software engineering techniques in different stages and aspects of software development; • practice in applying the theories, concepts and techniques acquired during lectures through the actual accomplishment of a guided case study project.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: *Professional/academic knowledge and skills* (a) apply software engineering techniques in the systems specifications and design stages of software projects; (b) apply software engineering techniques to real-life case study projects; (c) acquire concepts in software quality assurance standards and be able to develop skills and practices in quality software development; *Attributes for all-roundedness* (d) solve complex problems in groups and be able to communicate effectively through project presentations; (e) communicate in writing with technical documentation throughout the various stages of project development. **Alignment of Programme Outcomes:** Programme Outcome 1: Practice communication skill in discussion and project presentation and documentation. Programme Outcome 4: Think and reason critically on developing alternatives in problem solving and system development, and be able to design and test systems by applying related technologies. Programme Outcome 6: Follow closely the advancement in software design and testing and their impact to the software development projects.

<table>
<tr><td></td><td>Programme Outcome 7: Work together as a team in project design and development.</td></tr>
<tr><td><b>Subject Synopsis/ Indicative Syllabus</b></td><td>
<table>
<tr><th>Topic</th><th>Duration of Lectures</th></tr>
<tr><td><b>1. Software process</b>
Software process and process models; software reuse.</td><td>2.5</td></tr>
<tr><td><b>2. Specification and requirement analysis</b>
Event-based specification; model-based specification; requirements analysis; prototyping.</td><td>5</td></tr>
<tr><td><b>3. Software analysis and design</b>
System analysis and models; overview of software design process and strategies; function-oriented design; objected-oriented design.</td><td>12.5</td></tr>
<tr><td><b>4. Programming techniques and tools</b>
Programming style; fault avoidance and tolerance; exception handling; defensive programming; computer-aided software engineering; software development environment; features of programming languages.</td><td>2.5</td></tr>
<tr><td><b>5. Software verification and validation</b>
Testing techniques and tools; static analysis; formal proof; design and code reviews, inspection, walkthrough; software reliability; software safety.</td><td>7.5</td></tr>
<tr><td><b>6. Software metrics</b>
Complexity metrics; use of metrics for software monitoring and control; software quality assurance.</td><td>5</td></tr>
<tr><td><b>Total</b></td><td><b>35</b></td></tr>
</table>
<b>Case Study:</b>
Students will practise their skills in developing a group project representing a real-life application.</td></tr>
<tr><td><b>Teaching/Learning Methodology</b></td><td>Lectures focus on introduction and explanation of key concepts and techniques. Tutorial and lab sessions provide students opportunity to practice the techniques and tools presented in class. Assignments and project allow students to deepen their understanding of the concepts taught in class and apply the theory and techniques to software design and testing. Students will be encouraged to work in groups to share and present ideas, review other’s work, and develop teamwork skill.</td></tr>
</table>

Assessment Methods in Alignment with Intended Learning Outcomes

Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)					
		a	b	c	d	e	
1. Assignments	60%	✓		✓			
2. Lab exercises		✓	✓	✓			
3. Project			✓		✓	✓	
4. Mid-term		✓		✓			
5. Examination	40%	✓		✓			
Total	100 %						

Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:

Assignments, project and mid-term test act as a measure on the understandings of the students on the basic concepts of the software specification, design and testing.

In addition, project will be used to measure the understandings of the students about the current practice in software design and testing. The students can improve their presentation and communication skills through the project presentation, and practice team work. Students can also develop their analytic and problem solving skills.

Examination will be used as an overall measure of the understandings of the students on software development process, software specification, design and testing concepts and technologies.

Student Study Effort Required

Class contact:	
▪ Lecture	35 Hrs.
▪ Tutorial	14 Hrs.
Other student study effort:	
▪ Work on assignments and project; study related material/ team work	84 Hrs.
▪ Study for mid-term and examination	25 Hrs.
Total student study effort	158 Hrs.

Reading List and References

Textbooks:

1. Pressman, R., Software Engineering: A Practitioner's Approach, 6th Edition, McGraw-Hill, 2005.

Reference books:

1. Sommerville, I., Software Engineering, 9th Edition, Addison-Wesley, 2010.

	2. Booch, G., Object Oriented Analysis & Design with Applications, Second Edition, Addison-Wesley, 1994. 3. Jacobson, I., Booch, G. and Rumbaugh, J., The Unified Software Development Process, Addison-Wesley, 1999. 4. Pierre Bourque and Robert Dupuis, Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 2004. 5. Kathy Schwalbe, Information Technology Project Management, 6th Edition, Cengage Learning, 2009.