Subject Description Form

Subject Code	COMP3211
Subject Title	Software Engineering
Credit Value	3
Level	3
Pre-requisite / Co-requisite / Exclusion	**Pre-requisite**: COMP2011
Objectives	The objectives of this subject are to: • provide students with a general knowledge of the application of software engineering techniques in different stages and aspects of software development; and • provide students with practice in applying the theories, concepts and techniques acquired during lectures through the actual accomplishment of a 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) acquire concepts in software quality assurance and be able to test software applications; and (c) apply software engineering techniques to real-life case study projects. *Attributes for all-roundedness* (d) solve complex problems in groups and be able to communicate effectively through project presentations; and (e) communicate in writing with technical documentation throughout the various stages of project development.

Subject Synopsis/ Indicative Syllabus	Topic
	1. Software Process Software process and process models; component-based development; software reuse, CASE, agile development.
	2. Specification and Requirement Analysis Model-based specification; requirements analysis; prototyping.
	3. Software Analysis and Design System analysis and models; overview of software design process and strategies; function-oriented design; objected-oriented design
	4. Software Verification and Validation Testing techniques and tools; static analysis; design and code reviews, inspection
	5. Project Metrics Function point, line of code; COCOMO models; effort estimation.

Teaching/ Learning Methodology	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.

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
	Continuous Assessment	**60%**					
	1. Assignments		✓	✓			
	2. Lab Exercises		✓	✓	✓		
	3. Project				✓	✓	✓
	4. Mid-Term		✓	✓			
	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. 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 Expected	Class contact:	
	▪ Lecture	39 Hrs.
	▪ Tutorial/Lab	0 Hrs.
	Other student study effort:	
	▪ Assignments, Project and Self-study	66 Hrs.
	Total student study effort	105 Hrs.

Reading List and References	**Textbook:** 1. Sommerville, I., *Software Engineering*, 9th Edition, Addison-Wesley, 2010. **Reference Books:** 1. Pressman, R., *Software Engineering: A Practitioner's Approach*, 6th Edition, McGraw-Hill, 2006. 2. Booch, G., *Object Oriented Analysis & Design with Applications*, 2nd Edition, Addison-Wesley, 1994. 3. Jacobson, I., Booch, G. and Rumbaugh, J., *The Unified Software Development Process*, Addison-Wesley, 1999. 4. Bourque, Pierre and Dupuis, Robert, *Guide to the Software Engineering Body of Knowledge*, IEEE Computer Society, 2004. 5. Schwalbe, Kathy, *Information Technology Project Management*, 6th Edition, Cengage Learning, 2009.