

**SUMMARY OF INFORMATION ON EACH COURSE**

1.	Name of Course	Software Design	
2.	Course Code	TSE3151	
3.	Status of Course [Applies to (cohort) ]	Specialization Core	
4.	MQF Level/Stage Note : Certificate – MQF Level 3 Diploma – MQF Level 4 Bachelor – MQF Level 6 Masters – MQF Level 7 Doctoral – MQF Level 8	Bachelor – MQF Level 6	
5.	Version (State the date of the Senate approval – history of previous and current approval date)	Previous: June 2014 Current: June 2016	
6.	Pre-Requisite	TSE2101 Software Engineering Fundamentals	
7.	Name(s) of academic/teaching staff	Ho Chiung Ching Wan Ruslan bin W Yusoff Yeoh Eng Thiam	
8.	Semester and Year offered	Trimester 1 and 2 of every academic year	
9.	Objective of the course in the programme : To provide knowledge and expose students to the activities in software design covering design principles and concepts, qualities of good software design, negative design patterns, component level design and software architecture and framework		
10.	Justification for including the course in the programme : Justification for including the subject in the program: Software design will allow the continuous improvement of a software using software design patterns. It also gives a good overall view of a system via the use of architecture and frameworks		
11.	Course Learning Outcomes :	Domain	Level
	LO1. Discuss the qualities of good software design	Cognitive	1
	LO2. Analyze the quality of multiple software design based on key design principles and concepts	Cognitive	4
	LO3. Identify software design patterns in the construction of a software application	Cognitive	3
	LO4. Create and specify the software design for a medium sized software using a software requirement specification, an accepted program design methodology, appropriate notation and appropriately structured software	Cognitive	6

**SUMMARY OF INFORMATION ON EACH COURSE**

12.	Mapping of Learning Outcomes to Programme Outcomes :									
	Learning Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9
	LO1							X		
	LO2							X		
	LO3								X	
	LO4									X
13.	Assessment Methods and Types :									
	Method and Type	Description/Details						Percentage		
	Assignment	Group assignment						20%		
	Test	Written						20%		
	Quiz	Based on case study/tutorial/lab						10%		
	Final Exam	Written						50%		
14.	Mapping of assessment components to learning outcomes (LOs)									
	Assessment Components	LO1		LO2		LO3		LO4		
	Final Exam	X		X		X		X		
	Test	X		X						
	Assignment	X		X		X		X		
	Quiz	X		X		X		X		
15.	Details of Course									
	Topics			Mode of Delivery (eg. : Lecture, Tutorial, Workshop, Seminar, etc.) Indicate allocation of SLT (lecture, tutorial, lab) for each subtopic						
				Lecture (Hour)			Tutorial (Hour)			
	1.Introduction to Software Design What is software design Building model Transferring design knowledge Constraints of the software design process and product Recording software design decisions			2			0			

**SUMMARY OF INFORMATION ON EACH COURSE**

	<b>2. Fundamental design concepts and principles</b> From Software Requirements to Software Design Goals of software design Correctness Robustness Flexibility Reusability Efficiency	4	4
	<b>3. Software Design Patterns</b> What are design patterns Goals of design patterns Benefits of design patterns Using design patterns	6	6
	<b>4. Components Level Software Design</b> What are software components UML Component Notation Case study of component in use What components consists of Component life cycle Multicore, parallel and distributed component design	4	4
	<b>5. Software Design Qualities</b> Software Quality concept Assessing software design process & quality Quality attributes of the design product Contemporary software design challenges	6	6

**SUMMARY OF INFORMATION ON EACH COURSE**

	<b>Architecture and Frameworks</b> Meaning of software architecture Goals for architecture and modularization Modularization, cohesion and coupling Standard software architecture Meaning of Framework Framework Usage Goals for Framework Framework development and example	6	6																														
	<table><tr><th>Total Student Learning Time (SLT)</th><th>Face to Face / Guided Learning</th><th>Independent Learning</th></tr><tr><td>Lecture</td><td>28</td><td>28</td></tr><tr><td>Tutorials</td><td>26</td><td>26</td></tr><tr><td>Quiz</td><td>1</td><td>1</td></tr><tr><td>Presentation</td><td>0</td><td>0</td></tr><tr><td>Assignment</td><td>1</td><td>12</td></tr><tr><td>Mid Term Test</td><td>1</td><td>14</td></tr><tr><td>Final Exam</td><td>2</td><td>20</td></tr><tr><td>Sub Total</td><td>59</td><td>101</td></tr><tr><td>Total SLT</td><td colspan="2">160/40=4</td></tr></table>	Total Student Learning Time (SLT)	Face to Face / Guided Learning	Independent Learning	Lecture	28	28	Tutorials	26	26	Quiz	1	1	Presentation	0	0	Assignment	1	12	Mid Term Test	1	14	Final Exam	2	20	Sub Total	59	101	Total SLT	160/40=4			
Total Student Learning Time (SLT)	Face to Face / Guided Learning	Independent Learning																															
Lecture	28	28																															
Tutorials	26	26																															
Quiz	1	1																															
Presentation	0	0																															
Assignment	1	12																															
Mid Term Test	1	14																															
Final Exam	2	20																															
Sub Total	59	101																															
Total SLT	160/40=4																																
16.	Credit Value	4																															
17.	Reading Materials :																																
	Textbooks																																
	Hassan Gomma, Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, Cambridge University Press, 2011.																																
	Reference Material (including 'Statutes' for Law)																																
	<ul style="list-style-type: none"><li>Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra, Head First Design Patterns Publisher: O'Reilly Media, 2004.</li><li>Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, <i>Design Patterns: Elements of Reusable Object-Oriented Software</i>, Addison-Wesley, 1995 (Classic and famous seminal book for software design patterns.)</li><li>David Budgen, <i>Software Design (2nd Edition)</i>, Addison Wesley, 2003</li><li>Bernd Bruegge, Allen H. Dutoit, <i>Object Oriented Software Engineering Using UML, Patterns and Java (Second Edition)</i>, Prentice Hall, 2003</li><li>Eric J. Braude, <i>Software Design: From Programming to Architecture (First Edition)</i>, Wiley, 2003</li></ul>																																

**SUMMARY OF INFORMATION ON EACH COURSE**

Appendix (to be compiled when submitting the complete syllabus for the programme) :

1. Mission and Vision of the University and Faculty
2. Programme Objectives or Programme Educational Objectives
3. Programme Outcomes (POs)
4. Mapping of POs to the 8 MQF domain
5. Summary of the Bloom's Taxonomy's Domain Coverage in all the Los in the format below :

Subject	Learning Outcomes (please state the learning Outcomes)	Bloom's Taxonomy Domain		
		Affective	Cognitive	Psychomotor
TSE3151	LO1		1	
	LO2		4	
	LO3		3	
	LO4		6	

6. Summary of LO to PO measurement
7. Measurement and Tabulation of result for LO achievement
8. Measurement Tabulation of result for PO achievement