MULTIMEDIA UNIVERSITY

SUMMARY OF INFORMATION ON EACH COURSE/MODULE

1.	Name of Course/Module	Functional Programming
2.	Course Code	TCP2551
3.	Status of Subject	Elective Major
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 Degree – MQF Level 6
5.	Version (state the date of the last Senate approval)	Date of New Version : Year 2012
6.	Pre-Requisite	TMA1201 Discrete Structures & Probability
7.	Name(s) of academic/teaching staff	Prof. Yashwant Prasad Singh Assc. Prof Dr. C.K.Ho
8.	Semester and Year offered	Trimester 1 (Beta Level)
9.	Objective of the course/module in the programme :	
	This course introduces the functional programming paradigm and the implementation technology for functional programming languages. It aims to develop a broad understanding of the benefits of the functional programming style, together with an understanding of implementation issues that are relevant not only to functional languages but also to other systems that require automatic dynamic memory management.	
11.	Subject Learning Outcomes :	
	To be able to: • write programs in a functional style and be able to use the language to implement algorithms and data types to solve problems. • understand the basics of the lambda calculus and combinators and how they are used in the implementation of functional languages; • understand the main features of a modern lazy functional language; write non-trivial functional programs; understand the computation, synchronisation and memory management issues affecting the sequential and parallel implementation of lazy functional languages; • learn how to decompose and represent programming problems, how to compose the solutions into complete programs learn how to reason about the programs to ensure that they are correct	
13.	Synopsis:	
	Kursus ini memperkenalkan paradigma pengaturcaraan fungsi dan teknologi pelaksanaan untuk bahasa pengaturcaraan fungsi. Ia bertujuan untuk membina kefahaman tentang kelebihan gaya pengaturcaraan fungsi dan juga pemahaman terhadap isu pelaksanaan yang relevan kepada bukan sahaja bahasa fungsi bahkan terhadap sistem lain yang memerlukan pengurusan momri dinamik secara automatik.	
16.	Mapping of Subject to Programme Outcomes :	
	Programme Outcomes	**% of Contribution**
	Apply soft skills in work and career related activities	5
	To make use of fundamentals concepts and formulate best practices.	30

	Apply technical concepts and practices in specialized areas of Computer Science	20
	Analyze the requirements to address problems or opportunities faced by organizations	30
	Recognize and pursue continued life-long learning throughout their career	5
	Blend innovative mind and entrepreneurial skills	5
	Relate moral values and professional ethics to the practice of an ICT professional.	5

18. Assessment Methods and Types :

Method and Type	Description/Details	Percentage
Coursework:		
Midterm test		30%
Quizzes and Assignment		30%
Final Exams		40%

19. Details of Subject

Topics	Mode of Delivery	
	Lecture	Tutorial/Lab
Introduction Classification of Programming Languages; Distinctive Features of Functional Programming Languages; Principles of functional programming: expressions, evaluations, functions, and types	4	4
The Lambda Calculus and Combinators Reduction orders, strong normalisation Combinators - computationally complete sets	4	4
A Modern Functional Language Programming Environment-1 (Haskell-1) Programming Environment-2 (Haskell-2) Type definitions and built-in types: numbers, characters, strings, tuples and lists Recursion Pattern-Matching Higher-Order Functions User-Defined Types and type classes Recursive Programming Techniques	6	6
Data structures Binary trees, general trees. Use of trees for representing sets and symbolic data. Normal order reduction and lazy evaluation. Simple cost models for functional programs; time and space complexity.	6	6
Monad and Interaction Parsing Expression Parsing Expression Monad style Arithmetic expression parser	4	4
Programming GUIs	4	4
Total	**28**	**28**

23. Lab / Tutorial :

25.

Total Student Learning Time (SLT)	Face to Face (Hour)	Total Guided and Independent Learning
Lectures	28	28
Lab	28	28
Midterm test (1)	1	6
Assignment (1)	0	12
Quizzes	1	6
Lab test (1)	2	8
Final exam (1)	2	20

	SUBTOTAL	**60**	**100**
	Total SLT	**160/40 = 4**	
26.	Credit Value		4
27.	Reading Materials :		
	Textbook		Reference Materials
	Richard Bird, Introduction to Functional Programming using Haskell, Second Edition, Second Edition, Prentice-Hall International, 1998. This is more advanced than Thompson and best suited to students with a good background in mathematics. Simon Thompson, Haskell: The Craft of Functional Programming, Second Edition. Addison-Wesley, 1999.		1. Graham Hutton, Programming in Haskell, Cambridge University Press, 2007 2. Simon Thompson, Haskell: The Craft of Functional Programming, Addison-Wesley, Second Edition, 1999. 3. Paul Hudak, The Haskell School of Expression, Cambridge University Press, 2000. 4. "The Fun of Programming" edited by Jeremy Gibbons and Oege de Moor ("The Fun"). 5. George Springer and Daniel P. Friedman, “Scheme and the Art of Programming “, MIT Press, 1989. 6. Daniel P. Friedman and Matthias Felleisen, “The Little Schemer”, The MIT Press, 1995
30.	Appendix (to be compiled when submitting the complete syllabus for the programme) : 1. Mission and Vision of the University and Faculty 2. Mapping of Programme Objectives to Vision and Mission of Faculty and University 3. Mapping of Programme Outcome to Programme Objectives 4. Progarmme Objective and Outcomes (Measurement and Descriptions)		