MULTIMEDIA UNIVERSITY®

COURSE INFORMATION

1 .	**Name of Course**	Algorithm Design and Analysis
2 .	**Course Code**	TCP2101
3 .	**Type of Course** **(e.g. : Core, major, elective etc.)**	Specialization Core
4 .	**Synopsis**	The course aims to provide an understanding of performance of computer algorithms and data structures through the analysis of its space and time efficiency. The course also provide an introduction to common algorithm design paradigms.
5 .	**Version** **(State the date of theSenate's approval - previous and the current approval date)**	Current: January 2018 Previous: June 2016
6 .	**Name(s) of Academic Staff**	Ng Keng Hoong Neoh Kee Lin Angeline Pang
7 .	**Semester and Year Offered**	Trimester 2 (Gamma)
8 .	**Credit Value**	4
9 .	**Pre-Requisite**	TCP1201 Object-Oriented Programming and Data Structures

10 . **Objective of the course in the programme:**

- To introduce algorithm design techniques through the study of algorithms from various design paradigms.
- To introduce the techniques for analyzing algorithm efficiencies in respect of time and space complexity

11 . **Justification for including the course in the programme:**

To provide students with good analytical and design skills in computer algorithm.

12 .	**Course Learning Outcomes (CLO)**		**Domain**	**Level**
	CLO1:	Comprehend the complexity of polynomial and non-polynomial time algorithms	Cognitive	2
	CLO2:	Compare algorithms with respect to time and space requirements	Cognitive	4
	CLO3:	Select the best algorithms for solving problems	Cognitive	5
	CLO4:	Write efficient algorithms in terms of space and time complexities	Cognitive	6

13 . **Mapping of the Course Learning Outcomes to the Programme Learning Outcomes, Teaching Methods and Assessment:**

Course Learning Outcomes (CLO) (Must tally with CLOs in item 12)	**Programme Learning Outcomes (PLO)**												**Teaching Methods**	**Assessment Method**
	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12		
CLO1		✓	✓										Lecture/Practical	Assignment/Quiz/Test/Final Exam
CLO2		✓	✓										Lecture/Practical	Assignment/Quiz/Test/Final Exam
CLO3		✓	✓	✓									Lecture/Practical	Assignment/Quiz/Test/Final Exam
CLO4	✓	✓	✓	✓		✓							Lecture/Practical	Assignment/Test/Final Exam
Total	**1**	**4**	**4**	**2**		**1**							*Indicate the relevancy between the CLO and PLO by ticking "✓" the appropriate relevant box (This description must be read together with standards 2.1.2, 2.2.1, and 2.2.2 in Area 2 – pages 16 & 18 of COPPA 2.0)*	

14 . **Transferable Skills:**

Critical thinking, through the analysis of time and spatial efficiency of algorithms, assessed in Assignment, Quiz, Test, and Exam.

15 . **Distribution of Student Learning Time (SLT)**

	Course Content Outline	****CLO**	**Teaching and Learning Activities Guided Learning (F2F)***				**Guided Learning (NF2F)***	**Independent Learning (NF2F)***	**Total SLT**
			***L**	***T**	***P**	***O**			
1	**Basic Algorithmic Analysis** Proof technique, asymptotic analysis of upper and average complexity bounds (best, average, and worst case behaviours), big-Oh, big-Omega, and big-Theta notation, standard complexity classes, empirical measurements of performance, time and space trade-offs in algorithms, using recurrence relations to analyse recursive algorithms)		3		1			4	8
2	**Review of Basic Data Structures** Stack ADT, Queue ADT, List ADT, array structure, linked-list structure, Map ADT, Dictionary ADT, hash table, Set ADT, Union/Find structures		2		1			2	5
3	**Tree** General tree, tree implementation, Binary Tree ADT, tree traversal (preorder, inorder, postorder), binary search tree, red-black tree, Heap and Heap-sort, Priority Queue ADT		5		2			6	13
4	**Graphs** Graph ADT, adjacency matrix structure, adjacency list structure, graph traversal (Depth-First Search, Breadth-First Search), spanning tree, Euler cycles, maximum flow problem		5		3			6	14
5	**Divide and Conquer** Master Theorem, Strassen's matrix product algorithm, integer multiplication		5		4			7	16
6	**Sorting and Selection Algorithms** Selection sort, Merge-sort, Quick-sort, Radix sort, topological sorting, Quick-select		5		4			7	16
7	**Greedy Algorithms** Dijkstra's algorithm (shortest path problem), Kruskal's and Prim's algorithm (minimum spanning tree problem), Huffman codes, continuous-knapsack problem		5		4			7	16

	8	**Dynamic Programming** Matrix multiplication, long common subsequence problem, Floyd and Warshall’s algorithm, subset sums		3		2		7	4	16
	9	**Text Processing** Brute force, Rabin-Karp’s algorithm, Knuth-Morris-Pratt’s algorithm, Boyer-Moore’s algorithm		4		3			6	13
	10	**P and NP Problems** Polynomial time (P), nondeterministic algorithms and Nondeterministic Polynomial time (NP), Reducibility and NP-Completeness, NP-Complete problems: satisfiability (SAT), 3SAT, graph colouring, Hamiltonion cycles, travelling salesperson problem		3		0			3	6
									Total SLT	**123**

SUMMATIVE ASSESSMENT

1. Continuous Assessment	**Percentage %**	**Total SLT**	
Assignment	20%	10	
Quiz	20%	4	
Test	20%	5	
	Total SLT for Continuous Assessment	**19**	
2. Final Assessment	**Percentage %**	**Total SLT**	
		F2F	**ILT**
Final Exam	40%	2	16
	Total SLT for Final Assessment (F2F + NF2F)	**18**	
Grand Total	**100%**	**160**	

****Indicate the CLO based on the CLO’s numbering in Item 12.**
***L= Lecture, *T= Tutorial, *P= Practical, *O= Others, F2F*= Face to Face, NF2F*= Non Face to Face**

16 . **Identify Special Requirement to Deliver the Course (e.g., software, nursery, computer lab, simulation room):**
Latest C++ compiler (Code::Blocks) and Java compiler

17 . **Main References:**
Goodrich, M. T., & Tamassia, R. (2014). Algorithm Design and Applications. New York: John Wiley & Sons

18 . **Additional References:**
1. Alsuwaiyel, M. H. (2016). Algorithms Design Techniques and Analysis (Revised Edition). World Scientific Publishing.
2. Cormen, T.H., Leiserson, C.E., Rivest, R.L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press.
3. Sedgewick, R. & Wayne, K. (2016). Algorithms (4th ed.). Addison-Wesley.

Note:

▨	Cells shaded light grey contain formulas / fixed values. Edit these formulas only if needed.