Teaching Plan

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DATA STRUCTURE AND ALGORITHM

DITP 2113	SEMESTER I	SESI 2021/2022

DITP 2113 DATA STRUCTURES AND ALGORITHMS [3, 2, 2]

TYPE OF COURSE: K
EDITION: 2
UPDATED: 03-10-2021

1.0 LEARNING OUTCOMES

Upon completion this course, students will be able to:

i. Illustrate the algorithm design and performance for different abstract data type operation. (C3)
ii. Construct and apply suitable data structures and algorithm for an application that requires data structures. (P3)
iii. Demonstrate the data structures and algorithms in problem solving. (CS3, CTPS3)

2.0 SYNOPSIS

This course aims to develop students' knowledge in data structures and algorithms. The course begins with the introduction of concepts and techniques of structuring and operating on Abstract Data Types in problem solving. Followed by the discussion on the operations for maintaining common data structures. Students are exposed on how to recognize the associated algorithms' operations and complexity. Common sorting, searching and graph algorithms will be discussed and the complexity and comparisons among these various techniques will be studied.

3.0 PRE-REQUISITE

DITP 1113 Programming I, DITP 1123 Programming II

4.0 PRACTICAL

Microsoft Visual C++ will be used as the tool for practical session. Students will attend the laboratory sessions for constructing programs using various aspects of algorithm and abstract data type implementation that they have learnt.

5.0 REFERENCES

[1] Malik, D. S., "C++ Programming: Program Design Including Data Structures", 8th edition, Cengage Learning, 2018.

[2] Drozdek, A., "Data Structures and Algorithms in C++ 4th Edition", Cengage Learning, 2013.

[3] Malik, D. S., "Data Structures Using C++", 2nd edition, Cengage Learning, 2010.

[4] Y.Daniel Liang, "Introduction to Programming with C++",3rd Edition, Pearson, 2014.

[5] Mark A. Weiss, "Data Structures and Algorithm Analysis in C++",4th edition, Pearson, 2014.

6.0 COURSE IMPLEMENTATION

i. Lecture
   2 hours per week for 14 weeks (Total = 28 hours)

ii. Laboratory Activities
   2 hours per week for 14 weeks (Total = 28 hours)

7.0 COURSE EVALUATION

Assessment Method	LO 1/PO1	LO 2/PO2	LO 3/PO5	
Lab Assessment = 20%		LA (20%)		Scheme
Lab Test = 15%		LT (15%)		Scheme
Project = 15%			P (15%)	Rubric
Mid Term = 20%	MT (20%)			Scheme
Final = 30%	F (30%)			Scheme
Total	**50%**	**35%**	**15%**	

8.0 STUDENT LEARNING TIME (SLT)

		Guided Learning Time (hr)				Independant Learning(hr)								Assessment Time (hr)				
Minggu *Week*	CLO	L	T	P	O	L	T	P	O	F	T	A	O	F	T	A	O	SLT
W1	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W2	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W3	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W4	1,2	2	0	2	0	1	0	1	0	0	0	2	0	0	0	0.5	0	8.5
W5	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W6	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W7	1,2	2	0	2	0	1	0	1	0	0	4	0	0	0	1	0	0	11
W8	1,2	2	0	2	0	1	0	1	0	0	8	0	0	0	2	0	0	16
W9	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W10	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W11	1,2	2	0	2	0	1	0	1	0	0	0	2	0	0	0	0.5	0	8.5
W12	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W13	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W14	1,2	2	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	6
W15	1,3	0	0	0	0	0	0	0	0	10	0	0	4	2.5	0	0	1	17.5
Keseluruhan *Overall*		28	0	28	0	14	0	14	0	10	12	4	4	2.5	3	1	1	121.5
																	SLT Credit Equivalent	3.04

9.0 DETAILED SYLLABUS AND TEACHING PLAN

WEEK	CONTENT	REF	DELIVERY METHOD	NOTES
1,2	**CHAPTER 1:** **CLASS AND OBJECT** 1.1. Class and object concept. 1.2. Access modifier public, private and protected. 1.3. Constructor and destructor. 1.4. Inheritance – extends and implements (interface class). 1.5. Overloading and overriding.	[1,4]	Lecture	
1,2	1.1. Create classes to perform array, pointer, function and structure. 1.2. Build program code to apply overloading, inheritance and overriding.		Lab	
3	**CHAPTER 2:** **TEMPLATE, INTRODUCTION TO DATA STRUCTURES AND ALGORITHMS ANALYSIS.** 2.1. Class and function template. 2.2. Introduction to data structures and algorithms	[1,4]	Lecture	
3	2.1. Build template in a program. 2.2. Differentiate Class template and Function Template via code examples.		Lab	

4,5	**CHAPTER 3:** **ASYMPTOTIC NOTATION AND SEARCHING** 3.1. Algorithm analysis and big-O 3.2. Concept of Sequential Search and Binary Search. 3.3. Analyze the run-time analysis in searching algorithms. 3.4. Introduction to Searching. 3.5. Searching algorithms for Sequential Search and Binary Search.	[1,4]	Lecture	
4,5	3.1. Perform the Big-O and run-time analysis in program. 3.2. Create searching program using Sequential Search and Binary Search.		Lab	Lab Assessment 1
6	**CHAPTER 4:** **LIST** 4.1. Introduction to list and linear list. 4.2. Linear list operations – create, add, delete and print item. 4.3. Introduction to node and linked list. 4.4. Linked list add operations – add at front.	[1,4]	Lecture	
6	4.1. Build program to illustrate the linear list. 4.2. Create a node for linked list. 4.3. Construct program for linked list add operations.		Lab	
7	**CHAPTER 5:** **LIST (CONTINUE)** 5.1. Linked list operations – create and add node. 5.2. Delete and print node in linked list. 5.3. Doubly linked list and circular linked list.	[1,4]	Lecture	
7	5.1 Construct program for linked list and doubly linked list operations.		Lab	Lab Test
8	**CHAPTER 6:** **SORTING** 6.1. Apprehend the algorithms of Simple Sort, Bubble Sort, Insertion Sort and Quick Sort. 6.2. Algorithm Analysis on run time for each sorting algorithms.	[1,4]	Lecture	
8	6.1 Convert the algorithms of Simple Sort, Bubble Sort, Insertion Sort and Quick Sort into program code.		Lab	Mid Term Test
9	**CHAPTER 7:**	[1,4]	Lecture	

	STACK 7.1. Introduction to stack. 7.2. Stack operations - push, pop, top. 7.3. Discussion on the advantages of stack in applications. 7.4. Examine the algorithms of converting decimal number to binary and balancing bracket.			
9	7.1. Implement stack operations in programs. 7.2. Apply the algorithms of converting decimal number to binary and balancing bracket.		Lab	
10	**CHAPTER 8:** **STACK (CONTINUE)** 8.1. Continue examining the algorithms of converting infix to postfix and prefix notation, evaluating postfix expression and addition two integer numbers.	[1,4]	Lecture	
10	8.1 Apply converting infix to postfix and prefix notation, evaluating postfix expression and addition two integer numbers algorithms.		Lab	
11	**CHAPTER 9:** **QUEUE** 9.1. Introduction to linear queue. 9.2. Linear queue operations – enqueue and dequeue. 9.3. Introduction to linked list queue and the operations	[1,4]	Lecture	
11	9.1 Build programs related to linear and linked list queue.		Lab	Lab Assessment 2
12	**CHAPTER 10:** **QUEUE (CONTINUE)** 10.1 Algorithm evaluating prefix expression. 10.2 Circular queue concepts.	[1,4]	Lecture	
12	10.1 Translate the evaluating prefix expression algorithm into program code. 10.2 Build programs related to circular queue.		Lab	
13	**CHAPTER 11:** **TREE** 11.1 Introduction to general tree, binary tree and binary search tree. 11.2 Traversal binary tree and operation of binary search tree. 11.3 Rules to delete nod in binary search tree.	[1,4]	Lecture	

	11.4 Expression Tree – build and traversal (prefix, postfix and infix) 11.5 Balance tree – AVL tree concept. 11.6 Process of insertion and deletion node in AVL tree.			
13	11.1 Exercises and programs related to tree		Lab	
14	**CHAPTER 12:** **GRAPH AND HEAPS** 12.1 Introduction to graph. 12.2 Graph traversals and graph algorithm. 12.3 Introduction to Heaps. 12.4 Heaps structure. 12.5 Basic heaps algorithms.	[1,4]	Lecture	
14	12.1 Draw binary search tree. 12.2 Build tree program that concern to manipulate the tree operation. 12.3 Create programs to relate to Expression tree and AVL tree. 12.4 Draw graph operations. 12.5 Draw heaps operations.		Lab	Project Presentation
				Final Exam

10.0 MATRIX OF LEARNING OUTCOMES

SUBJECT vs PROGRAM OUTCOME (PO)

Subject	PROGRAM OUTCOME (PO)								
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9
DITP 2113	X	X			X				

LEARNING OUTCOME (LO)

LO1	llustrate the algorithm design and performance for different abstract data type operation. (C3)
LO2	Construct and apply suitable data structures and algorithm for an application that requires data structures. (P3)
LO3	Demonstrate the data structures and algorithms in problem solving. (CS3, CTPS3)

LEARNING OUTCOME (LO) vs PROGRAM OUTCOME (PO)

LO	PROGRAM OUTCOME (PO)								
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9
LO1	X								
LO2		X							
LO3					X				

SUBJECT vs SOFT SKILLS

Subject	SOFT SKILLS																							
	communication skill					critical thinking & problem solving					team work			lifelong learning			entrepreneurship skills			ethics & moral professionalism			leadership skills		
	CS 1	CS 2	CS 3	CS 4	CS 5	CT PS1	CT PS2	CT PS3	CT PS4	CT PS5	TS 1	TS 2	TS 3	LL 1	LL 2	LL 3	ES 1	ES 2	ES 3	EM 1	EM 2	EM 3	LS 1	LS 2	LS 3
DITP 2113			X					X																

LEARNING OUTCOME (LO) vs SOFT SKILLS

LO	SOFT SKILLS																							
	communication skill					critical thinking & problem solving					team work			lifelong learning			entrepreneurship skills			ethics & moral professionalism			leadership skills		
	CS 1	CS 2	CS 3	CS 4	CS 5	CT PS1	CT PS2	CT PS3	CT PS4	CT PS5	TS 1	TS 2	TS 3	LL 1	LL 2	LL 3	ES 1	ES 2	ES 3	EM 1	EM 2	EM 3	LS 1	LS 2	LS 3
LO1																								
LO2																								
LO3			X					X																

SUBJECT vs TAXONOMY

Subject	Taxonomy																	
	Affective					Cognitive						Psychomotor						
	A1	A2	A3	A4	A 5	C1	C2	C3	C4	C5	C6	P1	P2	P3	P4	P5	P6	P7
DITP 2113			X					X						X				

LEARNING OUTCOME (LO) vs TAXONOMY

LO	Taxonomy																	
	Affective					Cognitive						Psychomotor						
	A1	A2	A3	A4	A5	C1	C2	C3	C4	C5	C6	P1	P2	P3	P4	P5	P6	P7
LO1								X										
LO2														X				
LO3			X															