Subject Description Form

Subject Code	COMP 305					
Subject Title	Data Structures and Algorithms					
Credit Value	3					
Level	3					
Pre-requisite /	Pre-requisite: COMP201 (Nil for 61025)					
Exclusion	Co-requisite/Exclusion: Nil					
Objectives	• To provide knowledge in various data structures and algorithms.					
	• To introduce techniques for analyzing the efficiency of computer algorithms.					
Intended Learning	Upon completion of the subject, students will be able to:					
Outcomes	Professional/academic knowledge and skills					
	(a) understand the properties of various data structures;					
	(b) identify the strengths and weaknesses of different data structures;					
	(c) design and employ appropriate data structures for solving computing problems;					
	(d) possess the knowledge of various existing algorithms;					
	(e) analyze and compare the efficiency of algorithms;					
	(f) possess the ability to design efficient algorithms for solving computing problems;					
	<u>Attributes for all-roundedness</u>					
	(g) solve problems independently;					
	(h) think critically for improvement in solutions.					
	Alignment of Programme Outcomes:					
	Programme Outcome 1: This subject contributes to communicative effectively by having students working in small teams to discuss and present programming in the lab and solving data structure design problems.					
	Programme Outcome 2: This subject contributes to the global outlook by having students understand the use of different computer platforms for different applications and their uses.					
	Programme Outcome 4: This subject contributes to critical thinking through					

	tutorial and lab exercises as well as direct exchanges on novel uses of data structures and algorithms. They will also practice in written assignments, programming exercises, and potential projects.					
	Programme Outcome 5: This subject contributes to technical problem solving by initiating a wide variety of algorithm design and implementation skills through lab exercise and project with proper design and implementation.					
	Programme Outcome 7: This subject contributes to team work by employing a small group-based approach to lab problem solving, assignments and miniprojects.					
Subject Synopsis/ Indicative Syllabus	Topic 1. Introduction Types of algorithms; analysis of algorithms; data structures; abstract data types.					
	2. Analysis of algorithms Mathematical techniques; classification of algorithms and their efficiencies; average-case and worst-case analysis.					
	3. Data structures: representation and algorithms Linear structures: linked-lists, stacks, queues; tree structures: binary trees, balanced trees, <i>m</i> -way trees, tree traversals; other common data structures: priority queues, heaps.					
	4. Sorting and searching algorithms Quadratic-time algorithms: bubble sort, insertion sort, selection sort; optimal-time algorithms: quick sort, merge sort, heap sort; searching algorithms: sequential search, binary search, tree search, dictionary and hashing.					
	5. Graph algorithms Depth-first and breadth-first search; test for acyclicity; topological sorting.					
	6. Text processing and data compression Prefix and suffix; dictionary; run-length encoding; Huffman coding.					
	7. Selected advanced topics Advanced topics such as AVL trees, divide-and-conquer.					
	Laboratory Experiment: Use of different data structures					
Teaching/Learning Methodology	The teaching methodology is based on three main activities:					
	1. Lecture delivery					
	2. Interactive exchange with students in class					

	3. Laboratory exercises consisting of hands-on programming exercises and tests											
	4. Tutorial sessions in and/or outside the lecture and laboratory sessions											
	5.	5. Office hours questions, answers and clarification of material										
	6.	6. Discussion sessions with optional additional workshops, lectures and labs								nd		
	The learning methodology will be based on:											
	1.	1. Lecture notes										
	2.	2. Laboratory notes and programming exercises										
	3.	 Textbook material Additional reference material 										
	4.											
	5.	5. Web links to active tutorials and other presentation material										
	6. Group interactions and supervised discussion sessions.											
Assessment Methods in Alignment with Intended Learning Outcomes	Is Specific assessment % Intended s assessed (bject learning outcomes to be ease tick as appropriate)					
o uteomes				a	b	c	d	e	f	g	h	
	1. As	signments	30%		~	~	~	✓				
	2. Lab exercises				✓	~	✓	~				
	3. Mi	d-term	30%	~	~	~	~	~		✓		
	4. Ex	amination	40%	~	✓	~	√	~				
	Total 100 %											
	The assignment weights will be effectively distributed amongst the intended subject learning outcomes to nurture creative thinking, independence, teamwork, technical skills and a global perspective towards the technological base of this subject. Specifically, the assignments and the lab exercises are selected to develop the technical skills and knowledge to solve problems in computing and software development as well as to realize effective solutions, understand, evaluate and develop a critical perspective in the development of both small and large systems and integration of systems. Critical thinking, effective communication and a demonstrable global outlook will be incorporated at every level of exercises and mid-term examinations. The final examination accounts for a global and comprehensive understanding of the entire subject material and serves as the final checkpoint for the learning outcomes against technical skills and critical problem solving with respect to all components of programming and data structure design.											
Student Study	Class c	Class contact:										

Effort Expected	•	Lecture	39 Hrs.			
	•	Lab	13 Hrs.			
	Other student study effort:					
	•	Class participation	4 Hrs.			
	•	Course work: reading, discussions, homework	46 Hrs.			
	Total s	tudent study effort	102 Hrs.			
Reading List and References	Textb 1. 2. Referent 1.	 rk Allen Weiss, Data Structures and Algorithm Analysis in C++, ird Edition, Addison Wesley, 2007. nk M. Carrano, Data Abstraction & Problem Solving with C++: ills & Mirrors, Addison Wesley, 2007. Books: am Drozdek, Data Structures and Algorithms in C++, Broooks/Cole, 01. 				
	2.	Cormen, Leiserson, Rivest, Stein, Introduction to Algorithms, MIT Press, Second Edition, 2001.				