

Subject Description Form

Subject Code	COMP2013
Subject Title	Data Structures and Algorithms
Credit Value	3
Level	2
Pre-requisite/ Co-requisite/ Exclusion	Pre-requisite: COMP1011/COMP1012/ENG2002/LGT3109 & AMA1110/AMA1501/AMA2634 & AMA1751/AMA2111/AMA2131/AMA2512
Objectives	To introduce students to data structures and algorithms, and how to use them to solve computational problems.
Intended Learning Outcomes	<p>Upon completion of the subject, students will be able to:</p> <ul style="list-style-type: none"> (a) understand the properties of basic data structures, and their strengths and weaknesses; (b) understand how to analyse and compare the efficiency of algorithms; (c) possess the knowledge of common algorithms, and common techniques for designing algorithms; (d) understand the basic steps in designing and implementing efficient data structures and algorithms for solving computational problems efficiently in a high-level programming language; and (e) get familiar with critical thinking for computational efficiency.
Subject Synopsis/ Indicative Syllabus	<ol style="list-style-type: none"> 1. Analysis of algorithms <ul style="list-style-type: none"> • Asymptotic notations. • Efficiency analysis. 2. Data Structures <ul style="list-style-type: none"> • Linear structures: linked-lists, stacks, queues. • Tree structures: binary trees, balanced trees, tree traversals. 3. Searching and Sorting <ul style="list-style-type: none"> • Greedy sorting algorithms: bubble sort, insertion sort, selection sort. • Sorting algorithms based on divide and conquer: quicksort and mergesort. • Heapsort and related data structures: priority queues, heaps. • Hashing: hash functions, collision resolution. 4. Graphs <ul style="list-style-type: none"> • Definition and representation. • Depth-first search and breadth-first search.

	<p>5. Algorithmic Design Techniques</p> <ul style="list-style-type: none"> • Greedy and dynamic programming. • Divide-and-conquer. • Recursion. <p>6. Applications</p> <ul style="list-style-type: none"> • Practical program development using combination of multiple data structures and algorithms. 																																																											
<p>Teaching/Learning Methodology</p>	<p>The course material will be delivered as a combination of mass lectures and small group supervised tutorial and laboratory sessions.</p> <p>Lectures will provide the required knowledge while tutorial and laboratory sessions allow students to acquire hands-on experience on programming with different algorithms.</p> <p>Both written and programming assignments will be utilised in the course. Written assignments help students develop analysis and design skills, whereas programming assignments emphasise on implementation skills.</p>																																																											
<p>Assessment Methods in Alignment with Intended Learning Outcomes</p>	<table border="1" data-bbox="533 987 1391 1697"> <thead> <tr> <th rowspan="2">Specific assessment methods/tasks</th> <th rowspan="2">% weighting</th> <th colspan="5">Intended subject learning outcomes to be assessed</th> </tr> <tr> <th>a</th> <th>b</th> <th>c</th> <th>d</th> <th>e</th> </tr> </thead> <tbody> <tr> <td>Continuous Assessment</td> <td>60%</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>1. Written Assignments</td> <td>15%</td> <td>✓</td> <td>✓</td> <td>✓</td> <td>✓</td> <td>✓</td> </tr> <tr> <td>2. Programming Assignments</td> <td>20%</td> <td>✓</td> <td>✓</td> <td>✓</td> <td>✓</td> <td>✓</td> </tr> <tr> <td>3. Quizzes</td> <td>25%</td> <td>✓</td> <td>✓</td> <td>✓</td> <td></td> <td>✓</td> </tr> <tr> <td>Examination</td> <td>40%</td> <td>✓</td> <td>✓</td> <td>✓</td> <td></td> <td>✓</td> </tr> <tr> <td>Total</td> <td>100%</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>						Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed					a	b	c	d	e	Continuous Assessment	60%						1. Written Assignments	15%	✓	✓	✓	✓	✓	2. Programming Assignments	20%	✓	✓	✓	✓	✓	3. Quizzes	25%	✓	✓	✓		✓	Examination	40%	✓	✓	✓		✓	Total	100%					
Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed																																																										
		a	b	c	d	e																																																						
Continuous Assessment	60%																																																											
1. Written Assignments	15%	✓	✓	✓	✓	✓																																																						
2. Programming Assignments	20%	✓	✓	✓	✓	✓																																																						
3. Quizzes	25%	✓	✓	✓		✓																																																						
Examination	40%	✓	✓	✓		✓																																																						
Total	100%																																																											
<p>Student Study Effort Expected</p>	<p>Class contact:</p>																																																											
	<ul style="list-style-type: none"> ▪ Lecture 					26 Hrs.																																																						
	<ul style="list-style-type: none"> ▪ Tutorial/Lab 					13 Hrs.																																																						
	<p>Other student study effort:</p>																																																											
	<ul style="list-style-type: none"> ▪ Written Assignments, Programming Assignments, Quizzes, Self-study 					101 Hrs.																																																						

	Total student study effort	140 Hrs.
Reading List and References	<ol style="list-style-type: none"> 1. Goodrich, Michael T., Tamassia, Roberto, and Goldwasser, Michael H., <i>Data Structures and Algorithms in Java</i>, 6th Edition, Wiley, 2014. 2. Goodrich, Michael T., Tamassia, Roberto, and Goldwasser, Michael H., <i>Data structures and algorithms in Python</i>, Wiley, 2013. 3. Sedgewick, Robert and Wayne, Kevin, <i>Algorithms</i>, 4th Edition, Addison-Wesley, 2011. 4. Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. and Stein, Clifford, <i>Introduction to Algorithms</i>, 4th Edition, MIT Press, 2022. 5. Frank M. Carrano, <i>Data Abstraction & Problem Solving with C++: Walls & Mirrors</i>, 7th Edition, Pearson, 2017 	