

Subject Description Form

Subject Code	COMP 2011
Subject Title	Data Structures
Credit Value	3
Level	2
Pre-requisite/ Co-requisite/ Exclusion	COMP 1011
Objectives	<p>The objectives of this subject are:</p> <ol style="list-style-type: none"> 1. To introduce students to basic concepts of data structures and algorithms 2. To teach students to apply simple data structures and algorithms in developing computer programs
Intended Learning Outcomes	<p>Upon completion of the subject, students will be able to:</p> <ol style="list-style-type: none"> (a) understand the properties of basic data structures; (b) identify the strengths and weaknesses of different data structures; (c) possess the knowledge of various common algorithms; (d) design and employ appropriate data structures and algorithms for developing computer applications; (e) think critically for improvement in the solutions.
Subject Synopsis/ Indicative Syllabus	<ol style="list-style-type: none"> 1. Programming and algorithms Computer algorithms; types of algorithms; data structures; abstract data types. 2. Data structures: representation and algorithms Linear structures: linked-lists, stacks, queues; tree structures: binary trees, balanced trees, tree traversals; other common data structures: priority queues, heaps. 3. Sorting Common sorting algorithms: bubble sort, insertion sort, selection sort; optimal-time sorting algorithms: quick sort, merge sort, heap sort. 4. Searching Common searching algorithms: sequential search, binary search; advanced searching algorithms: tree search, dictionary and hashing. 5. Applications Practical program development using combination of various

	data structures and algorithms, e.g. friends-book; efficiency of the various approaches.																																															
Teaching/Learning Methodology	The course material will be delivered as a combination of mass lectures and small group supervised tutorial and laboratory sessions. Lectures will provide the required knowledge while tutorials and laboratory sessions allow students to acquire hands-on experience on programming with different algorithms. Programming project provides students with a chance to integrate their knowledge on applying appropriate data structures and algorithms to solve practical problems.																																															
Assessment Methods in Alignment with Intended Learning Outcomes	<table border="1"> <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 (Please tick as appropriate)</th> </tr> <tr> <th>a</th> <th>b</th> <th>c</th> <th>d</th> <th>e</th> </tr> </thead> <tbody> <tr> <td>1. Laboratory Exercises</td> <td>20</td> <td>X</td> <td></td> <td>X</td> <td>X</td> <td></td> </tr> <tr> <td>2. Programming Project</td> <td>20</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> </tr> <tr> <td>3. Test</td> <td>20</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td></td> </tr> <tr> <td>4. Examination</td> <td>40</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> </tr> <tr> <td>Total</td> <td>100 %</td> <td colspan="5"></td> </tr> </tbody> </table>	Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)					a	b	c	d	e	1. Laboratory Exercises	20	X		X	X		2. Programming Project	20	X	X	X	X	X	3. Test	20	X	X	X	X		4. Examination	40	X	X	X	X	X	Total	100 %					
Specific assessment methods/tasks	% weighting			Intended subject learning outcomes to be assessed (Please tick as appropriate)																																												
		a	b	c	d	e																																										
1. Laboratory Exercises	20	X		X	X																																											
2. Programming Project	20	X	X	X	X	X																																										
3. Test	20	X	X	X	X																																											
4. Examination	40	X	X	X	X	X																																										
Total	100 %																																															
Student study effort expected	<table border="1"> <tr> <td colspan="2">Class Contact:</td> </tr> <tr> <td>Lecture</td> <td>39 hours</td> </tr> <tr> <td>Tutorial/Lab</td> <td>13 hours</td> </tr> <tr> <td colspan="2">Other student study effort:</td> </tr> <tr> <td>Assignments, Quizzes, Projects, Self-study</td> <td>55 hours</td> </tr> <tr> <td>Total student study effort</td> <td>107 hours</td> </tr> </table>	Class Contact:		Lecture	39 hours	Tutorial/Lab	13 hours	Other student study effort:		Assignments, Quizzes, Projects, Self-study	55 hours	Total student study effort	107 hours																																			
Class Contact:																																																
Lecture	39 hours																																															
Tutorial/Lab	13 hours																																															
Other student study effort:																																																
Assignments, Quizzes, Projects, Self-study	55 hours																																															
Total student study effort	107 hours																																															
Reading list and references	<ol style="list-style-type: none"> 1. Frank M. Carrano, Data Abstraction & Problem Solving with C++: Walls & Mirrors, Addison Wesley, 2007. 2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Third Edition, Addison Wesley, 2006. 3. Mark Allen Weiss, Data Structures and Algorithm Analysis in C, Second Edition, Addison Wesley, 1997. 3. Goodrich, M.T. and Tamassia, R., Data Structures and Algorithms in Java, Fifth Edition, John Wiley, 2010. 																																															