Subject Description Form

Subject Code	DSAI2201
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
Objectives	To introduce students to data structures and algorithms, and how to use them to solve computational problems.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: (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	**1. Analysis of algorithms** • Asymptotic notations. • Efficiency analysis. **2. Data Structures** • Linear structures: linked-lists, stacks, queues. • Tree structures: binary trees, balanced trees, tree traversals. **3. Searching and Sorting** • 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** • Definition and representation. • Depth-first search and breadth-first search. **5. Algorithmic Design Techniques** • Greedy and dynamic programming.

Subject Synopsis/ Indicative Syllabus (Cont'd)

- Divide-and-conquer.
- Recursion.

6. **Applications**

- Practical program development using combination of multiple data structures and algorithms.

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 tutorial and laboratory sessions allow students to acquire hands-on experience on programming with different algorithms.

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.

Assessment Methods in Alignment with Intended Learning Outcomes

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%					

Student Study Effort Expected

Class contact:	
▪ Lecture/Tutorial/Lab	39 Hrs.
Other student study effort:	
▪ Written Assignments, Programming Assignments, Quizzes, Self-study	101 Hrs.
Total student study effort	140 Hrs.

Reading List and References	1. Goodrich, Michael T., Tamassia, Roberto, and Goldwasser, Michael H., *Data Structures and Algorithms in Java*, 6th Edition, Wiley, 2014. 2. Goodrich, Michael T., Tamassia, Roberto, and Goldwasser, Michael H., Data structures and algorithms in Python, Wiley, 2013. 3. Sedgewick, Robert and Wayne, Kevin, *Algorithms*, 4th Edition, Addison-Wesley, 2011. 4. Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. and Stein, Clifford, *Introduction to Algorithms*, 4th Edition, MIT Press, 2022. 5. Frank M. Carrano, Data Abstraction & Problem Solving with C++: Walls & Mirrors, 7th Edition, Pearson, 2017