Subject Description Form

Subject Code	COMP327
Subject Title	Data Structures and Algorithms II
Credit Value	3
Level	3
Pre-requisite / Co-requisite/ Exclusion	Pre-requisite: COMP201, COMP305 Co-requisite/Exclusion: Nil
Objectives	• To provide students with in-depth knowledge on the key elements in computer programming. • To introduce and practice advanced algorithms and programming techniques necessary for developing sophisticated computer application programs.
Intended Learning Outcomes	Upon completion of the subject, students will be able to: *Professional/academic knowledge and skills* (a) understand the properties of advanced data structures; (b) design algorithms and employ appropriate advanced data structures for solving computing problems efficiently; (c) analyze and compare the efficiency of algorithms; (d) design and implement efficient algorithms for solving computing problems in a high-level object-oriented programming language (e.g., C++ or Java); *Attributes for all-roundedness* (e) solve problems independently; (f) think critically for improvement in solutions. **Alignment of Programme Outcomes:** Programme Outcome 1: This subject contributes to communicate 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 algorithmic techniques for different applications.

<table>
<tr><td></td><td>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 and programming exercises.

Programme Outcome 5: This subject contributes to technical problem solving by initiating a wide variety of algorithm design and implementation skills through assignments and lab exercises 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.</td></tr>
<tr><td>Subject Synopsis/ Indicative Syllabus</td><td>
<table>
<tr><th>Topic</th></tr>
<tr><td>1. Analysis of algorithms
Mathematical techniques; big-O notation; efficiency analysis; recurring relations.</td></tr>
<tr><td>2. Advanced algorithmic design techniques
Dynamic programming, divide-and-conquer, branch-and-bound, greedy algorithm, amortized analysis.</td></tr>
<tr><td>3. Advanced data structures
B-trees, B+ trees, heaps, graphs.</td></tr>
<tr><td>4. Graph algorithms
Minimum spanning trees, shortest paths, maximum flow.</td></tr>
<tr><td>5. Computational geometry algorithms
Spatial range searching, indexing of spatial objects, convex hull, closest pairs</td></tr>
<tr><td>6. Advanced topics in algorithms
Advanced topics such as text searching, NP-completeness, approximation algorithms.</td></tr>
</table>

Laboratory Experiment: Use of advanced data structures and algorithmic techniques in programming</td></tr>
<tr><td>Teaching/Learning Methodology</td><td>Lectures provide students the main concepts of the topic, together with comprehensive examples for easy understanding.

Tutorials and lab sessions offer an opportunity to students for practicing their algorithmic analysis, design, and implementation techniques.

Both written and programming assignments will be utilized in the course. Written assignments help students develop analysis and design skills, whereas programming assignments emphasize on implementation skills.</td></tr>
</table>

Assessment Methods in Alignment with Intended Learning Outcomes

Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)					
		a	b	c	d	e	f
1. Assignments	60%	✓	✓	✓	✓		
2. Lab exercises		✓	✓	✓	✓		
3. Mid-term / Tests		✓	✓	✓		✓	✓
4. Examination	40%	✓	✓	✓		✓	✓
Total	100 %						

Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:

All four items are relevant to the assessment of the use of algorithms advanced data structures for problem solving, as well as their efficiency analysis (for items a, b, c).

In addition, programming exercises in assignments and lab sessions are used to assess implementation skills (for item d); whereas the mid-term / tests and the examination are used to assess independent problem solving and critical thinking skills (for items e, f).

Student Study Effort Expected

Class contact:	
▪ Lecture	26 Hrs.
▪ Tutorial/Lab	13 Hrs.
Other student study effort:	
▪ Assignments (written and programming)	66 Hrs.
Total student study effort	105 Hrs.

Reading List and References

Textbooks:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein Introduction to Algorithms, Third Edition, MIT Press, 2009.

Reference Books:

1. M.T. Goodrich, and R. Tamassia, Data Structures and Algorithms in Java, Third Edition, John Wiley, 2005.
2. Frank M. Carrano, Data Abstraction & Problem Solving with C++: Walls & Mirrors, Addison Wesley, 2007.