## Subject Description Form

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>COMP2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject Title</td>
<td>Data Structures</td>
</tr>
<tr>
<td>Credit Value</td>
<td>3</td>
</tr>
<tr>
<td>Level</td>
<td>2</td>
</tr>
</tbody>
</table>

### Pre-requisite / Co-requisite / Exclusion

Pre-requisite: COMP1011

### Objectives

The objectives of this subject are to:

- introduce students to basic concepts of data structures and algorithms; and
- teach students to apply simple data structures and algorithms in developing computer programs.

### Intended Learning Outcomes

Upon completion of the subject, students will be able to:

- understand the properties of basic data structures;
- identify the strengths and weaknesses of different data structures;
- possess the knowledge of various common algorithms;
- design and employ appropriate data structures and algorithms for developing computer applications; and
- think critically for improvement in the solutions.

### Subject Synopsis/Indicative Syllabus

#### Topic

1. **Programming and Algorithms**
   
   Computer algorithms; types of algorithms; data structures; and abstract data types.

2. **Data Structures: Representation and Algorithms**
   
   Linear structures: linked-lists, stacks, queues; tree structures: binary trees, balanced trees, tree traversals; and other common data structures: priority queues, heaps.

3. **Sorting**
   
   Basic sorting algorithms: bubble sort, insertion sort, selection sort; and advanced sorting algorithms: quicksort, mergesort, heapsort.

4. **Searching**
   
   Common searching algorithms: sequential search, binary search; and 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; and 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.

<table>
<thead>
<tr>
<th>Assessment Methods in Alignment with Intended Learning Outcomes</th>
<th>Specific assessment methods/tasks</th>
<th>% weighting</th>
<th>Intended subject learning outcomes to be assessed (Please tick as appropriate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Assessment</td>
<td>60%</td>
<td></td>
<td>a b c d e</td>
</tr>
<tr>
<td>1. Laboratory Exercises</td>
<td>20%</td>
<td>✔</td>
<td>✔ ✔ ✔</td>
</tr>
<tr>
<td>2. Programming Project</td>
<td>20%</td>
<td>✔ ✔ ✔ ✔ ✔</td>
<td></td>
</tr>
<tr>
<td>3. Test</td>
<td>20%</td>
<td>✔ ✔ ✔ ✔ ✔</td>
<td></td>
</tr>
<tr>
<td>Examination</td>
<td>40%</td>
<td>✔ ✔ ✔ ✔ ✔ ✔</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Student Study Effort Expected**

Class contact:

- Lecture
  - 39 Hrs.
- Tutorial/Lab
  - 13 Hrs.

Other student study effort:

- Assignments, Quizzes, Projects, Self-study
  - 55 Hrs.

Total student study effort

<table>
<thead>
<tr>
<th></th>
<th>107 Hrs.</th>
</tr>
</thead>
</table>

**Reading List and References**

**Reference Books:**

