# Subject Description Form

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>COMP 2011</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>
<tr>
<td>Pre-requisite/ Co-requisite/ Exclusion</td>
<td>COMP 1011</td>
</tr>
</tbody>
</table>

## Objectives

The objectives of this subject are:

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

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

(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

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>
<thead>
<tr>
<th>Specific assessment methods/tasks</th>
<th>% weight</th>
<th>Intended subject learning outcomes to be assessed (Please tick as appropriate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Laboratory Exercises</td>
<td>20</td>
<td>a X X X</td>
</tr>
<tr>
<td>2. Programming Project</td>
<td>20</td>
<td>X X X X X</td>
</tr>
<tr>
<td>3. Test</td>
<td>20</td>
<td>X X X X</td>
</tr>
<tr>
<td>4. Examination</td>
<td>40</td>
<td>X X X X X</td>
</tr>
<tr>
<td>Total</td>
<td>100 %</td>
<td></td>
</tr>
</tbody>
</table>

**Student study effort expected**

- **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**