ET2006 – Data Structures and Algorithms

<table>
<tr><td><b>Subject Code</b></td><td>ET2006</td><td><b>Subject Title</b></td><td colspan="4">Data Structures and Algorithms I</td></tr>
<tr><td><b>Credits</b></td><td>4.0</td><td rowspan="2"><b>Total Hours</b></td><td><b>Lectures</b></td><td>45 h</td><td rowspan="2"><b>Pre-Requisites</b></td><td rowspan="2">ET1001</td></tr>
<tr><td><b>GPA/NGPA</b></td><td>GPA</td><td><b>Lab/Assignment</b></td><td>30 h</td></tr>
</table>

Aim: To provide knowledge of basic data structures and algorithmic design paradigms used to manipulate data structures for common functions in software programming..

Learning Outcomes:

On the completion of this module, the student will be able to:

1. Select and implement common data structures for a given situation
2. Implement appropriate algorithms to manipulate data structures
3. Analyze the performance and complexity of basic algorithms
4. Use data structures and synthesize efficient algorithms to solve real-life problems.

LO 1.	**Select and implement common data structures for a given situation [12 h]** Basic data structures and their properties:- linear (arrays, linked-lists, stacks, queues) and non-linear (trees and graphs); Operations on data structures:- insertion, deletion, sorting, searching, traversing; Heap abstract data type and heap operations; Dictionaries and hash tables
LO 2.	**Implement appropriate algorithms to manipulate data structures [12 h]** Characteristics of an algorithm, algorithm creation techniques and metrics of algorithms; Recursion:- tail and non-tail, direct and indirect; Algorithms for stacks and queues:- first in first out (FIFO) and last in first out (LIFO) Sorting algorithms:- internal and external sorting, simple sorting algorithms (bubble, selection, insertion sort), and advanced sorting algorithms (merge, quick, bucket, shell, heap sort); Searching algorithms:- linear (sequential) search and binary search (binary search trees)
LO 3.	**Analyze the performance and complexity of basic algorithms [9 h]** Factors considered in analyzing algorithms, and growth of functions; Asymptotic notations: Ω, o and ω; Time complexity and asymptotic complexity analysis; Efficiency of different searching and sorting algorithms
LO 4.	**Use data structures and synthesize efficient algorithms to solve real-life problems [12 h]** Solving real-life recursion problems:- factorials, Fibonacci numbers, Tower of Hanoi; Divide and conquer techniques; Tree and graph traversal techniques:- breadth-first and depth-first, handling cycles and vertices; Tree traversal path orders:- pre-order, in-order, and post-order Graph traversal algorithms:- Bellman-Ford shortest-path and Dijkstra shortest path algorithm

Practical Work: [3×3 h]

1. Implementation of advanced data structures and advanced sorting algorithms
2. Representing a given problem using graphs, creating low charts, and writing pseudo code of algorithms
3. Analyzing the time complexity of a given algorithm

Assessment:

a. Assignment on LO 1 and LO 2	6th week	10%
b. Assignment on LO 3	9th week	10%
c. Assignment on LO 4	12th week	10%
d. End Semester Written Examination: 3 h		70%

Rec. Bks.	1. **Data Structures and Algorithms in Java**; by Adam Drozdek (3E) **Publisher:** Cengage Learning Asia. 2008; **ISBN:** 978-9814239233 2. **Data Structures and Algorithms in Java**; by Michael Goodrich and Roberto Tamassia (5E) **Publisher:** Wiley. 2010; **ISBN:** 978-0470383261 3. **Introduction to Algorithms**; by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein (3E) **Publisher:** The MIT Press. 2009: **ISBN:** 978-0262033848