Data Structures and Algorithms (IT605)

Components of course: Lectures and Labs

Mode of delivery: Live lectures as per institute timetable, additional recordings, for the lectures. For the labs I will assign programming assignments and ask for submissions of code on a suitable platform.

Evaluation: There will be one insemester exam (40%), one end semester exam (40%) and lab assignments (20%).

Course content (will be pruned dynamically in order to adhere to the institute’s guideline of reduced duration for this semester).

Abstract Data Types: Abstraction, ADT, Bags, Iterators, Application

Arrays: Array Structure, The Python list, 2D arrays, Matrix abstract data types, Application

Sets and Maps: Sets, Maps, Multi-dimensional arrays, Application

Algorithm analysis: Complexity analysis, Python list evaluation, Amortized cost, Set ADT evaluation, Application

Searching and sorting: Linear search, binary search, Bubble sort, Selection sort, Insertion sort, Working with sorted list.

Linked Structures: Singly linked list, Ways to build linked list, The bag ADT, Sparse matrix, Polynomials, Doubly linked list, circular linked list, multi linked list, Complex iterators.

Stacks: The stack ADT, Implementing stack, Applications.

Queues: The queue ADT, Implementing queue, Priority queues, Applications.

Recursion: Recursive functions, Properties of recursion, How recursion works, Recursive applications.

Hash Tables: Hashing, Separate chaining, Hash functions, The Hash-map abstract data type, Application

Tries: Standard (handling keys, searching, Deletion), Compressed, Suffix.

Advanced sorting: Merge sort, Quick sort, Radix sort, Sorting linked list.

Binary trees: The tree structure, The binary tree, Expression trees, Heaps, Heapsort, Applications
Search trees: The binary search tree, Search tree iterators, AVL trees, B-Trees, 2-3-4 Trees

References:

1. Data Structures and Algorithms - Aho, Hopcroft and Ullman [Addison-Wesley, 1999]
3. Introduction to Algorithms - Cormen, Leiserson, Rivest and Stein [PHI, 2010].