Data Structure and Algorithm

BEG273CO

Year: II Semester: III

Teaching Schedule Hours/Week			Examination Scheme				
Theory	Tutorial	Practical	Internal Assessment		Final		Total
			Theory	Practical	Theory	Practical	
3	-	3	20	50	80	-	150

Course Objective: To understand the fundamental concept of data structure. On the completion of this course the student will be able to design data structure and implement it using programming language.

Course Contents:

1.0 Introduction to data structure (2 Hrs)
- 1.1 Concept of data structure and its uses
- 1.2 Abstract data type (ADT): definition and importance
- 1.3 Implementation of data structure
- 1.4 Introduction and application of Big O notation

2.0 The Stack (2 Hrs)
- 2.1 Stack as ADT
- 2.2 Operation in stack and stack implementation
- 2.3 Application: evaluation of infix, postfix, and prefix expression

3.0 Queue (3 hrs)
- 3.1 Queue as an ADT, queue Implementation
- 3.2 Operation in queue: enque and deque
- 3.3 Linear and circular queue, and their application
- 3.4 Priority queue: definition and application

4.0 List (2 hrs)
- 4.1 Definition
- 4.2 Static and Dynamic list structure
- 4.3 Array Implementation of Lists, Stacks, and Queues as continuous list

5.0 Linked Lists (5 Hrs)
- 5.1 Definition
- 5.2 Linked list as an ADT
- 5.3 Implementation
- 5.4 Operation in linked list: node insertion, deletion, insertion and deletion after and before nodes
- 5.5 Applications of linked stack and queue
- 5.6 Doubly linked list and its applications
- 5.7 Circular linked list

6.0 Recursion (4 Hrs)
- 6.1 Recursion and principle of recursion
- 6.2 Need and importance of recursion
- 6.3 Recursion and iteration algorithm, Converting recursion to iteration
- 6.4 TOH and fibonacci sequences and recursion
- 6.5 Applications of recursion

7.0 Trees (6 Hrs)

- 7.1 Tree concept
- 7.2 Basic operation in tree: insertion, deletion and search
- 7.3 Tree height, depth, and level
- 7.4 Binary tree traversals (pre-order, post-order and in-order)
- 7.5 AVL balance trees, balancing algorithm
- 7.6 Huffman tree and its application

8.0 Sorting (6 Hrs)

- 8.1 Definition
- 8.2 Types of sort: internal and external sort
- 8.3 Insertion and selection sort, exchange sort
- 8.4 Quick sort and merge sort
- 8.5 Shell sort
- 8.6 Binary sort
- 8.7 Heap and heap sort as priority queue
- 8.8 Efficiency of sorting

9.0 Searching (7 Hrs)

- 9.1 Definition of searching
- 9.2 Searching technique
- 9.3 Essential of search
- 9.4 Types of search: Sequential, binary, tree, general search tree
- 9.5 Hashing: Hash function and Hash tables, Collision resolution technique
- 9.6 Efficiency comparisons of different search technique
  - 9.6.1 Big OH, Big Sigma, and Big Omega Notation
  - 9.6.2 Calculation of O(--) for a simple program

10.0 Graphs (8 Hrs)

- 10.1 Definition of graph
- 10.2 Representation and applications
- 10.3 Graphs as an ADT
- 10.4 Transitive closure
- 10.5 Graphs types, graph traversal and spanning forests
- 10.6 Kruskal's and shortest path algorithm

Practicals: The practicals should cover all the above chapters of this course in a high-level programming language.

Reference Books:

1. Y. Langsam, M. J. Augenstein & A. M. Tanenbaum, "Data Structures using C and C++", PHI
2. G. W. Rowe, "Introduction to Data Structure and Algorithms with C and C++", PHI
3. R. L. Kruse, B. P Leung, C. L. Tondo, "Data Structure and Program Design in C", PHI