**Data Structures**

**Programme:** B.Tech (CSE) **Year :** Second **Semester :**

**Course:** Core **Credits :** 4 **Hours :** 40 hours (Theory)

**Course Context and Overview (100 words):**

* 1. To allow to assess how the choice of data structures and algorithm design methods impacts the performance of programs
  2. To choose the appropriate data structure and algorithm design method for a specified application.
  3. To learn the systematic way of solving problems, various methods of organizing large amounts of data.
  4. To solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, heaps, tournament trees, binary search trees, and graphs and writing programs for these solutions.

**Prerequisites Courses:**

Computer Programming

**Textbook references (IEEE format):**

**Text Book:**

1. Aaron M. Tenenbaum, Y. Langsam, Moshe J. Augenstein. *Data Structures Using C*. PHI Learning (2009).
2. Aho A.V., J.E. Hop croft, J.D. Ullman. *Data Structures and algorithms*. Addison Wesley.
3. Mark Allen Weiss. *Data Structures and Algorithm Analysis in C++*. (2nd Edition)

**Reference books:**

1. Robert L. Kruse, Bruce P. Leung, Clovis L. Tondo. *Data Structures and Program Design In C (2nd Edition).* Prentice Hall, 1996.
2. Horowitz and Sahni. Data Structure in C++ , Glagotia.
3. Ellis Horowitz, Sartaj Sahni. *Fundamentals of Data Structures*. Computer Science Press.
4. Niklaus Wirth. *Algorithms + Data Structures = Programs*. Prentice-Hall Series in Automatic Computation.
5. Sartaj Sahni. *Data Structures, Algorithms, and Applications in C++.* TMH.

**Course outcomes (COs):**

|  |
| --- |
| **The Outcomes of this Course are** |
| **CO1:** To describe the usage of various data structures |
| **CO2:** To explain the operations for maintaining common data structures |
| **CO3:** To understand basic algorithms such as heap sort, graph traversal-based, sorting, AVL trees, and hashing |
| **CO4:** To implement different data structures and related algorithms |
| **CO5:** To choose the appropriate data structure to solve a programming problem |

**Course Topics**

|  |  |  |
| --- | --- | --- |
| **Contents** | **Lecture Hours** | |
| **UNIT – 1 Complexity Analysis** | | 2 |
| Time and Space complexity of algorithms, asymptotic analysis, big O and other notations, importance of efficient algorithms, program performance measurement, data structures and algorithms. | |
| **UNIT –2 Linear Lists** | | 10 |
| Abstract data type, sequential and linked representations, comparison of insertion, deletion and search operations for sequential and linked representation, linked lists, doubly linked lists, circular lists, Sorting (Insertion sort, selection sort, Bubble sort, Merge Sort, Quick Sort), applications of lists in bin sort, radix sort, sparse tables and polynomial operations. | |
| **UNIT-3 Stacks and Queues** | |  |
| Abstract data types, sequential and linked implementations of Stack and Queue , Stack applications such as parenthesis matching, towers of Hanoi, Expression conversion, Expression evaluation, multiple stacks; Dqueue ; Circular Queue | | 6 |
| **UNIT-4 Hashing** | | 2 |
| Search efficiency in lists and skip lists, hashing as a search structure, hash table, collision avoidance, linear open addressing, chains, uses of hash tables in text compression | |
| **UNIT-5 Trees** | | 5 |
| Binary trees and their properties, terminology, sequential and linked implementations, tree traversal methods and algorithms, heaps as priority queues, heap implementation, insertion and deletion operations, heap-sort, heaps in Huffman coding | |
| **UNIT-6 Search Trees** | | 6 |
| Binary search trees, search efficiency, insertion and deletion operations, importance of balancing, AVL trees, searching insertion and deletions in AVL trees, comparison with AVL trees, search insert and delete operations. | |
| **UNIT-7** **Multi-way Trees** | | 4 |
| Issues in large dictionaries, m-way search trees, B- trees, search insert and delete operations, height of B-tree, 2-3 trees, sets and multisets. | |
| **UNIT-8** **Graphs** | | 5 |
| Definition, terminology, directed and undirected graphs, properties, connectivity in graphs, applications, implementation – adjacency matrix and linked adjacency chains, graph traversal – breadth first and depth first, spanning trees. | |

**Evaluation Methods:**

|  |  |
| --- | --- |
| **Item** | **Weightage** |
| Mid Term 1 | 25 |
| Quiz + Assignment | 25 |
| End Term | 50 |

**Prepared By:**

**Last Update: 20.4.2015**