UE17CS202:
DATA STRUCTURES (4-0-0-0-4)

Course Objectives:
The objective(s) of this course is to,

- Enable the learner with the concepts of recursion and linear data structures viz., Linked Lists, Stacks and Queues.
- Enable the learner with the concepts of non-linear data structures viz., Graphs, Trees, Heaps, Trie and Hashing.
- Hone the learner such that they obtain the ability to compare different implementations of data structures and recognize the advantages and disadvantages of different implementations.
- Inculcate in the learner, the aspects of choosing the appropriate data structure and algorithm design method for a specified application and with the usage of standard libraries.

Course Outcomes:
At the end of the course, the student will be able to:

- Implement fundamental data structures viz., Lists, Stacks, Queues, Linked Lists, Binary Trees from first principles.
- Demonstrate the use of appropriate data structures for a given problem.
- Design and implement solutions to basic practical problems using customized data structures.
- Develop quick and foolproof solutions to practical problems using abstract data types.

Course Content:

1. **Data Structures Overview**: Recursion, Pointers, Programming Practices. **Lists**: Definition, Create, Insert, Delete, Update, Traverse and Position-based Operations, Linked List and Array Implementations, Concatenate, Merge, and Reverse Lists, Doubly-Linked List Implementation and Operations, Circular Lists and Multi-List, Applications of Lists.
2. **Stacks**: Definition, Operations, Implementation using Linked-List and Arrays, Applications of Stacks – Postfix Conversion and Expression Evaluation, Parentheses Balancing. **Queues**: Definition, Operations, Implementation, Applications, Circular Queue, Dequeue.
3. **Graphs:** Definition, Complete Graphs, Regular Graphs, Paths, Connectivity, Euler and Hamilton Graphs, Representation of Graphs - Adjacency/ Cost Matrix, Adjacency Lists, Traversal of Graphs. **Trees**: General Tree Representation, Traversals, Applications. **Binary Trees**: Definition, Properties, Implementation, Traversals, Applications.
4. **Binary Search Tree**: Definition, Implementation, Search, Insert, Delete Operations, Building and Evaluating Binary Expression Tree. **Heap Tree**: Implementation, Insert, Delete, FindMin Operations, Priority Queue using Arrays and Heap.
5. **Tries:** Definition, Implementation, Applications. **Hashing**: Hash Table, Hash Functions, Collision Handling by Open Addressing, Chaining.

Pre-requisite Courses: UE17CS151 – Problem Solving with C.

Reference Book(s):

1. "Data Structures and Program Design in C", Robert Kruse, C L Tondo, Bruce Leung, Shashi Mogalla, PHI, 2nd Edition, 2015.
2. "Data Structures Using C and C++", Tanenbaum, Langsam, Augenstein, Pearson, 2nd Edition, 2015.