

## Data Structure and Algorithms Lab

### Lab Program 4

1. Design and implement a menu-driven Java program to construct and manipulate an expression tree for a given arithmetic expression. The program should allow the user to build the tree from a valid infix or postfix expression, where internal nodes represent operators and leaf nodes represent operands. Once the expression tree is constructed, the program should support evaluating the expression represented by the tree and displaying it in different traversal orders to obtain prefix, infix, and postfix forms. It should also provide functionality to check whether the tree is empty, handle invalid expressions with appropriate messages, and display the tree structure clearly. The implementation should demonstrate how expression trees can be used for parsing and evaluating arithmetic expressions efficiently in compilers and interpreters.
2. Design and implement a menu-driven Java program to perform operations on a multi-way search tree suitable for indexing large amounts of data. The tree should allow multiple keys to be stored in a single node and should maintain balance automatically during insertions and deletions by redistributing or merging keys when required. The program should support inserting new keys, deleting existing keys, and searching for a given key efficiently. It should also provide options to traverse and display the contents of the tree in sorted order and in level-wise structure to visualize the hierarchy. The implementation must ensure that the tree remains balanced after every update so that search, insert, and delete operations can always be performed in logarithmic time. Proper messages should be displayed for unsuccessful search or delete operations, and the program should be tested with a set of sample keys to demonstrate how the structure grows and shrinks dynamically.