# Data Structure and Algorithm

**3rd Graded Lab**



**SESSION: Spring 2025**

# Faculty of Information Technology UCP Lahore, Pakistan

**Graded Lab – 3**

**Lab Topic**:

AVL Tree and its basic operations

**Lab Objective:**

* Understand how AVL works and maintain order.
* Practice implementing insert, search, and delete operations.
* Understand and apply recursive in-order traversal for alphabetical listing.
* Recognize how separating traversal logic (inOrderTraversal) supports modular and reusable code design.

## Instructions:

* Indent your code
* Comment your code
* Use meaningful variable names
* Plan your code carefully on a piece of paper before you implement it.

**Graded Lab 4**

## Lab Scenario:

### 

You are developing a Book Cataloging System for a digital library that stores book titles in alphabetical order. To maintain fast and efficient access as the library grows, your system will use an AVL Tree — a self-balancing Binary Search Tree.

### AVL Trees maintain balance by performing rotations during insertions and deletions, ensuring that no path becomes disproportionately deep. This makes the system ideal for frequent operations such as adding, removing, or searching for books.

### Objectives

Complete the AVL Tree-based Book Catalog system by implementing:

* **Insertion** of book titles while maintaining AVL balance
* **In-order traversal** to display titles in alphabetical order
* **Search** functionality to find if a specific book exists
* **Deletion** of books while preserving tree balance
* Utility methods:
  + Count of total books
  + Height of the AVL Tree

**Provided Code Include**

• Node structure with name and height

• Rotation methods:

leftRotate()

rightRotate()

• Utility methods:

height()

getBalance()

**Tasks for you**

Implement boolean findBook(String title)

* Check if a book exists in the AVL Tree.
* Return true if found, false otherwise.

Implement void deleteBook(String title)

* Remove a book from the AVL Tree.
* Ensure balance is maintained using provided rotations.

Implement the following methods as an optional task:

* int countBooks() – Returns total number of books.
* int getTreeHeight() – Returns the current height of the tree.
* void printTreeStructure() – Displays the tree structure using indentation/text format.