Assignment 3:
Optimizing Library Book Management Using Binary Search Tree (BST) and AVL Tree

CLO-2: Use non-linear data structures to solve computing problems.

Scenario:

Imagine you are tasked with designing a book management system for a library. The library has the following requirements:

Efficient Search:

Books are identified by unique ISBN numbers and prices, and users frequently search for books based on either of these attributes.

Balance Management:

The library often adds or removes books. The data structure must remain balanced, with the AVL tree allowing balancing based on price.

Visual Representation:

The librarian needs to see the tree structure at different stages to understand its balance.

Tasks:

1. BST Implementation

Write a C++ program to implement a Binary Search Tree (BST) for the library.

Include functions for:

1. Insertion of books (by ISBN and price).
2. Searching for a book by ISBN.
3. Searching for books by price range (e.g., find all books between $10 and $50).
4. Deletion of a book by ISBN.
5. Displaying the tree structure (in-order, pre-order, post-order traversals).

2. AVL Tree Implementation

Extend the BST to implement an AVL Tree with balancing based on price.

Ensure that after every insertion or deletion:

The tree remains balanced based on the book price.

Implement functions for single and double rotations (left and right).

Searching by price range should work efficiently due to AVL balancing.

3. Visualization

Provide visual diagrams showing the BST and AVL tree at the following stages:

After initial insertion of a set of books.

After a deletion operation.

After balancing (in AVL tree based on price).

Use a tool like Google Drawings, Canva, or any graph visualization library to create tree diagrams.