Practical 14

Aim : Write a Program to Build BST

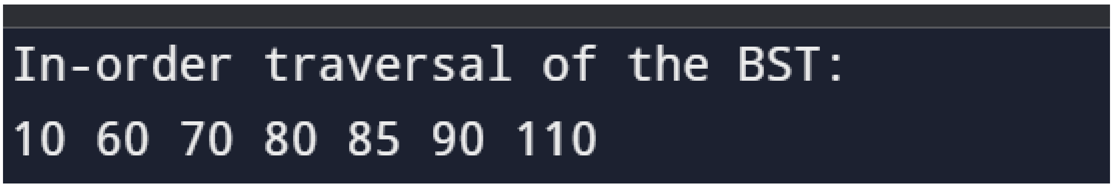
Algorithm :

- This Java code defines a Binary Search Tree (BST) class and a Solution class with methods to
- insert nodes into the BST and perform an inorder traversal. Here are the algorithms in short:
- BST Class:
  - Data: An integer representing the data in the node.
  - Left: A reference to the left child node in the BST.
  - Right: A reference to the right child node in the BST.
- Insert Method:
  - Insert a new node with the given value val into the BST.
  - If the root is null, create a new node with the value val.
  - If the value is greater than the root's data, recursively insert into the right
- subtree.
  - If the value is less than or equal to the root's data, recursively insert into the left
- subtree.
- Inorder Traversal Method:
  - Perform an inorder traversal of the BST (left subtree, root, right subtree).
  - If the root is null, return.
  - Recursively traverse the left subtree, then print the current node's data, and
- finally recursively traverse the right subtree.
- These algorithms work together to insert nodes into the BST in the correct order and print out
- the data in sorted order using an inorder traversal.

CODE:-

```
public class BST {
  int data;
  BST left;
  BST right;


  public BST(int data) {
    this.data = data;
  }


  public static void main(String[] args) {
    BST root = null;
    root = insert(root, 80);
    root = insert(root, 60);
    root = insert(root, 90);
    root = insert(root, 10);
    root = insert(root, 70);
    root = insert(root, 85);
    root = insert(root, 110);

    System.out.println("In-order traversal of the BST:");
    inorder(root);
}

public static BST insert(BST root, int val) {
    if (root == null) {
        return new BST(val);
    }
    if (root.data < val) {
        root.right = insert(root.right, val);
    } else {
        root.left = insert(root.left, val);
    }
    return root;
}

public static void inorder(BST root) {
        if (root == null) {
            return;
        }
        inorder(root.left);
        System.out.print(root.data + " ");
        inorder(root.right);
    }
}
```

Output :-



```
In-order traversal of the BST:
10 60 70 80 85 90 110
```