**WIA1002/WIB1002/WXES1117 Data Structures**

**Lab 12: BST**

1. Create a generic TreeNode class that accepts the following:
   1. A generic element variable
   2. Two references, respectively called left and right
   3. A default constructor with no initial value
   4. A constructor that accepts one element variable, assigned to itself
2. A generic BST class that extends the Comparable with the following methods
   1. A root variable of type TreeNode<E>
   2. An integer size variable initialized to 0
   3. public int search(E e) //returns 1 if element is found
   4. public String insert(E e) //returns the inserted element or ‘Duplicates not inserted’ if a

// duplicated value is inserted

* 1. public boolean delete(E e) //returns true when element is deleted
  2. protected void inorder(TreeNode<E> root) //inorder traversal
  3. protected void postorder(TreeNode<E> root) //postorder traversal
  4. protected void preorder(TreeNode<E> root) //preorder traversal
  5. public int getSize() //returns size
  6. public TreeNode<E> getRoot() //returns root
  7. public String rootElement() //returns element of the root
  8. public void clear() //remove all elements from tree

1. Write a test program called TestBST that :
   1. Creates a char object for the BST class
   2. Inserts the following item in the given order: a,b,c
   3. Display the tree by respectively invoking(calling) the inorder(), postorder() and preorder()
   4. Display root
   5. Display size
   6. Search for ‘b’
   7. Inserts ‘b’
   8. Inserts ‘d’
   9. Deletes c
2. Using stack/queue, implement the inorder traversal using a non-recursive approach.