



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: Summer 2022, B.Sc. in CSE (DAY)

LAB REPORT NO # 06

Course Title: Data Structure Lab

Course Code: CSE 106

Section: CSE 213 - DA (PC)

Lab Experiment Name(s):

- Implement a BST and traverse the tree using Pre-order, in-order and Post-order (by taking user choice) Traversal using linked list

Student Details

Name	ID
Md. Shahidul Islam Prodhan	213902017

Lab Date: 17 August 2022

Submission Date: 23 August 2022

Course Teacher's Name: Ms Farhana Akter Sunny, Senior Lecturer.

[For Teacher's use only: **Don't write anything inside this box**]

Lab Report Status

Marks:	Signature:
Comments:	Date:

1. TITLE OF THE LAB EXPERIMENT

Write a C program to Implement a BST and traverse the tree using Pre-order, inorder and Post-order (by taking user choice) Traversal using linked list

2. OBJECTIVES

Implement insertion of BST and Traverse them

3. PROCEDURE/ ANALYSIS / DESIGN

Algorithm of insertion <pre>if(node==NULL){ return newnode(key); } if(node->data > key) node->left = insert(node->left,key); else if(node->data < key) node->right = insert(node->right,key); return node;</pre>	Algorithm of traverse In-order <pre>if (root != NULL) inorder(root->left); printf("%d \n", root->data); inorder(root->right);</pre>
Algorithm of traverse pre-order <pre>if (root != NULL) { printf("%d \n", root->data); inorder(root->left); inorder(root->right); }</pre>	Algorithm of traverse Post-order <pre>if (root != NULL) inorder(root->left); inorder(root->right); printf("%d \n", root->data);</pre>

4. IMPLEMENTATION & TEST RESULT

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct node {
5      int item;
6      struct node* left;
7      struct node* right;
8  };
9
10
11 void inorderTraversal(struct node* root) {
12     if (root == NULL) return;
13     inorderTraversal(root->left);
14     printf("%d ->", root->item);
15     inorderTraversal(root->right);
16 }
17
18
19 void preorderTraversal(struct node* root) {
20     if (root == NULL) return;
21     printf("%d ->", root->item);
22     preorderTraversal(root->left);
23     preorderTraversal(root->right);
24 }
25
26
27 void postorderTraversal(struct node* root) {
28     if (root == NULL) return;
29     postorderTraversal(root->left);
30     postorderTraversal(root->right);
31     printf("%d ->", root->item);
32 }
33
34 struct node* createNode(value) {
35     struct node* newNode = malloc(sizeof(struct node));
36     newNode->item = value;
37     newNode->left = NULL;
38     newNode->right = NULL;
39
40     return newNode;
41 }
42 struct node* insertLeft(struct node* root, int value) {
43     root->left = createNode(value);
44     return root->left;
45 }
46 struct node* insertRight(struct node* root, int value) {
47     root->right = createNode(value);
48     return root->right;
49 }
50 int main() {
51     struct node* root = createNode(1);
52     insertLeft(root, 12);
53     insertRight(root, 9);
54
55     insertLeft(root->left, 5);
56     insertRight(root->left, 6);
57
58     printf("Inorder traversal \n");
59     inorderTraversal(root);
60
61     printf("\nPreorder traversal \n");
62     preorderTraversal(root);
63
64     printf("\nPostorder traversal \n");
65     postorderTraversal(root);
66 }
```

4. IMPLEMENTATION & TEST RESULT

Output

```
/tmp/9U3GdR80nb.o
```

```
Inorder traversal
```

```
5 ->12 ->6 ->1 ->9 ->
```

```
Preorder traversal
```

```
1 ->12 ->5 ->6 ->9 ->
```

```
Postorder traversal
```

```
5 ->6 ->12 ->9 ->1 ->
```

6. ANALYSIS AND DISCUSSION

- 1) This problem is solved by using c program. In this program we implement insertion of Binary Search Tree. And Traverse in In-Order, Pre-order, Post-Order.

7. SUMMARY

1. We complete this problem in c programming language.