



Department of IT and Computer Science
Pak-Austria Fachhochschule: Institute of Applied Sciences
and Technology, Haripur, Pakistan

COMP-201L Data Structures and Algorithms Lab

Lab Report 10

Class: **Computer Science**
Name: **Yaseen Ejaz Ahmed**
Registration No.: **B20F0283CS014**
Semester: **Third**
Submitted to: **Engr. Rafi Ullah**

Instructor Signature

Lab No. 10

Binary Trees

Objectives:

To understand:

- Binary Trees
- Implementation
- Construction

Tools/Software Required:

C++ Compiler

Introduction:

A binary tree is a hierarchical data structure in which each node has at most two children. The child nodes are called the left child and the right child.

To start with, let's describe the linked list representation of a binary tree in which each node has three fields:

- Pointer to store the address of the left child
- Data element
- Pointer to store the address of the right child

Lab Tasks:

Lab Task 01: Construct a Binary Tree (5-1-3-6-7-4-2)

Code:

```
#include <iostream>
using namespace std;

//5-1-3-6-7-4-2

class Node
{
    public:
        int data;
        Node *root;
        Node *left, *right;

    Node* Insert(int data)
    {
        Node* temp = new Node;
        temp->data = data;
        temp->left = temp->right = NULL;
        return temp;
    }
};

int main()
{
    Node n;
    n.root = n.Insert(5);
    n.root->left = n.Insert(1);
    n.root->left->right = n.Insert(3);
    n.root->left->right->left = n.Insert(2);
    n.root->left->right->right = n.Insert(4);
    n.root->right = n.Insert(6);
    n.root->right->right = n.Insert(7);
```

```

/*      5
      /  \
     1    6
      \    \
      3     7
      /\
     2  4
*/

}

```

Results & Observations:

In this lab, we have learnt about the basic implementation and construction of binary trees. Binary trees are a type of sorted trees. New nodes are compared with present nodes to see if the new number is smaller or larger. Using this algorithm, we can construct a binary tree.