# 

# Department of Software Engineering

**CS 250: Data Structures and Algorithms**

**Class: BESE-7AB**

**Lab 10: Binary Search Tree**

**CLO1: Understand the fundamentals of data structures and algorithms.**

**Date: December 15th, 2017**

**Time: 9:00 am -12:00pm, 2:00pm – 5:00pm**

# Instructor: Dr. Muhammad Shahzad

# Lab 10: Implementation of Binary Search Tree

**Introduction**

This lab is based on the implementation of Binary Search tree and its functions.

**Objectives**

The objectives of this lab are the following:

* Become familiar with implementation of binary search trees
* Study some statistics of binary search trees
* Write simple applications using binary search tree

**Tools/Software Requirement**

Visual Studio 2012 or gcc or g++

**Description**

In computer science, a binary search tree (BST), which may sometimes also be called an ordered or sorted binary tree, is a node-based binary tree data structure which has the following properties:

* The left sub-tree of a node contains only nodes with keys less than the node's key.
* The right sub-tree of a node contains only nodes with keys greater than the node's key.
* Both the left and right sub-trees must also be binary search trees.
* There must be no duplicate nodes.

In this lab, you will expand implement binary search tree, study some statistical properties of BST and write a simple application using the BST.

Here is a template of how your class/structure looks like.

classBST\_Node{

Template data;

BST\_Node \* LeftChild;

BST\_Node\* RightChild;

};

**Lab Task**

You are required to upload the lab tasks on LMS and the name of that tasks must be in this format

FullName\_reg#\_task#.cpp

Remember to comment your code properly. Inappropriate or no comment will results in deduction of marks.

**You are already provided with the basic implementation of Binary Search Tree. Run the code and you are required to do the following.**

**Tasks**

Implement the following operations of Binary Search Tree ADT

1. Write main method that calls insert on user’s request. After the addition the resultant tree must be printed
2. Modify the Insert function so that it handles the duplicate values. The left sub-tree of any node shall contain items that are less than the root. Items that are equal to or greater than the root must go to the right.
3. Write a function to find the minimum value in a BST
4. Write a function to calculate the height of the BST

**Deliverable**

Students are required to upload the lab task on LMS before the deadline.