|  |  |  |  |
| --- | --- | --- | --- |
| animatedLOGO | **Assignment No. 01 (Graded) SEMESTER Fall 2018 CS301- Data Structures** | | **Total Marks: 20**  **Due Date: 10 /12 /18** |
| ***Instructions***  **Please read the following instructions carefully before solving & submitting assignment:**  **It should be clear that your assignment will not get any credit (zero marks) if:**   * **The assignment is submitted after due date.** * **The submitted assignment is other than .cpp file.** * **The submitted assignment does NOT open or file is corrupted.** * **The assignment is copied (from other student or ditto copy from handouts or internet).**   ***Uploading instructions***   * For clarity and simplicity, you are required to Upload/Submit only **.cpp file.**   ***Objective***  The objective of this assignment is:   * To make you understand the different traversal forms of a Binary Search Tree data structure.   **For any query about the assignment, contact at** [**cs301@vu.edu.pk**](mailto:cs301@vu.edu.pk)  **GOOD LUCK** | | | |
|  | | **Marks: 20** | |

Binary Search tree is the most efficient data structure for solving different problems. Construction of BST (Binary Search Tree) depends on the order in which data is inserted into the tree. For building a Binary Search Tree, data should be inserted into a tree in a way that nodes with smaller data values appears on left side and larger node values appears on right side of the root node.

**Write a menu based system program in C++ that will allow you to:**

|  |  |
| --- | --- |
| Enter Employee data in BST | **buildBST()** |
| Post order traversal of all the Employee data | **postOrder()** |
| Show data in ascending order | **asscendingOrder()** |

Take the following attributes of an Employee from the user:

**empId, empNname, empSalary**

You have to implement the above scenario using BST on the basis of Employee Id. i.e. if the Id of the Employee is lesser than the root node then enter the next Employee data in the left child otherwise in right child.

Note:

* BST will implement using Linked List.
* Program will not allow entering duplicate employee id.
* Design the program in a way that the user will be able to enter maximum 10 records.
* Take empId, empName, empSalary from the user. At least 4 students record should be already entered (hard coded records).

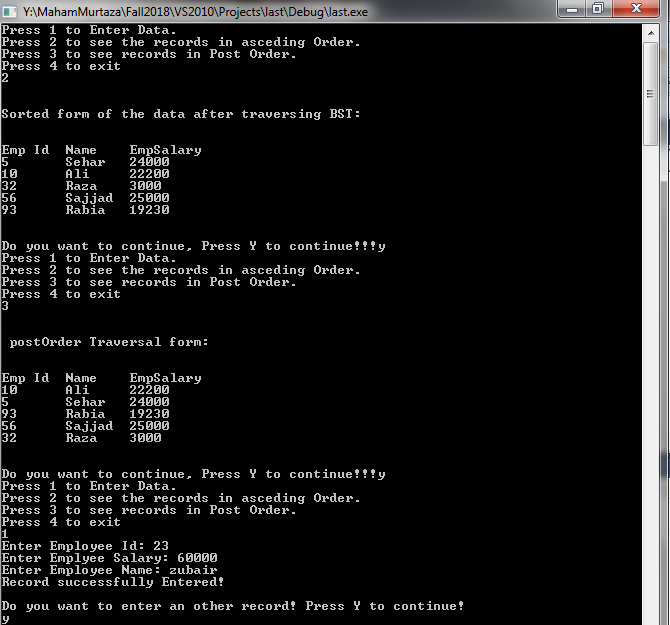
You will use following already entered Employees data (hard coded records).

|  |  |  |
| --- | --- | --- |
| **Emp Id** | **Name** | **EmpSalary** |
| 32 | Raza | 3000 |
| 56 | Sajjad | 25000 |
| 93 | Rabia | 19230 |
| 5 | Sehar | 24000 |
| 10 | Ali | 22200 |

**Solution Guidelines:**

* You will use **buildBST()** method to build Binary Search Tree from the above given data.
* Use the **ascendingOredr()** method to show the output in ascending order. With respect to **empId**.
* Use **postOrder()** method to traverse the records in post-order.

**Sample output 1:**



**Sample output 2:**

