Binary Search Tree

Source Code:

```
/*
Name : Robin Singh
number : 1261
unit : 06
program : Binary Search Trees
*/
#include<iostrea
m>
#include<conio.h>

using namespace std;

/*Node
Template*/ class
BSTNode
{
        public
        :       int data;
                BSTNode
                *right;
                BSTNode *left;
};

/* Binary tree
template*/ class BST
{
        BSTNode
        *root; int cnt;
                public:
                        BST
                        ()
                        {       root=NUL
                                L; cnt=0;

                        }

                void Insert(int
                x); void
                Display(); void
                FindMax(); void
                FindMin();
                void CountNodes();
                void Preorder(BSTNode
                *t); void
                Postorder(BSTNode *t);
                void Inorder(BSTNode *t);
```

```
};
/* Functions */
void BST :: Insert(int x)
{
        BSTNode *t=new
        BSTNode(); t->data=x;
        t-
        >right=NULL;
        t->left=NULL;

        if(root==NULL)
        {
                root=t;
                cnt++;
                return
                ;
        }
        BSTNode
        *tmp=root;
        BSTNode
        *prev=NULL;

        while(tmp!=NULL)
        {
                prev=tmp;
                if(t->data < tmp->data)
                {
                        tmp=tmp->left;
                }
                else if(t->data > tmp->data)
                {
                        tmp=tmp->right;
                }
                els
                e
                {       cout<<"Duplicate!!"<<en
                        dl; return;

                }

        }//End of while


                if(t->data < prev->data)
                {
                        prev->left=t;
                }
                else
```

```
{
```

```
                        prev->right=t;
                }
                cnt++;



}//end of Insert
/* Display function
*/ void BST ::
Display()
{
        cout<<"Preorder :
        "; Preorder(root);
        cout<<"end" <<
        endl;
        cout<<"Postorder :
        "; Postorder(root);
        cout<<"end"<<
        endl;
        cout<<"Inorder : ";
        Inorder(root);
        cout<<"end"<<endl
        ;
}
void BST :: Inorder(BSTNode *t)
{
        if(t)
        {
                Inorder(t->left);
                cout<<t->data << "-
                "; Inorder(t->right);
        }
}

void BST :: Preorder(BSTNode *t)
{
        if(t)
        {
                cout<<t->data << "-
                "; Preorder(t->left);
                Preorder(t->right);
        }
}

void BST :: Postorder(BSTNode *t)
{
        if(t)
        {
```

```
                Postorder(t->left);
                Postorder(t-
                >right); cout<<t-
                >data<<"-";
        }
}



/* Find Minimum */
void BST ::
FindMin()
{
        if(root==NULL)
        {
                cout<<"Empty
                tree"; return;
        }
        BSTNode *tmp =
        root; BSTNode *prev
        = NULL;
        while(tmp!=NULL)
        {
                prev=tmp;
                tmp=tmp-
                >left;

        }
        cout<<"Minimum Node is "<<prev->data;
}
/* Find Maximum */
void BST ::
FindMax()
{

        if(root==NULL)
        {
                cout<<"Empty
                tree"; return;
        }
        BSTNode *tmp =
        root; BSTNode *prev
        = NULL;
        while(tmp!=NULL)
        {
                prev=tmp;
                tmp=tmp-
```

```
            >right;

        }
        cout<<"Maximum Node is "<<prev->data;
```

```
}

void BST :: CountNodes()
{
	cout << "Number Of the node in the BST : "<<cnt;
}



/* Menu */
int main()
{
	BST b1;
	int num,ch;

	while(1)
	{
		system("cls");

			cout<<"*********Binary Search
			Tree(BST)*********"<<endl<<endl; cout<<"1.Insert Node in the
			binary search tree : \n"; cout<<"2.Display The BST : \n";
			cout<<"3.Find the maximum node :
			\n"; cout<<"4.Find the minimum node
			: \n"; cout<<"5.Count the node in the
			BST : \n"; cout<<"6.Exit \n";
			cout<<"Enter The choice
			: "; cin>>ch;

			switch(ch)
			{
				case 1:
					cout<<"Enter the number to Insert in
					BST: "; cin>>num;
					b1.Insert(num
					); getch();
					break;

				case 2:
					b1.Display();
					getch()
					;
					break;

				case 3:
```

```
                    b1.FindMax();
                    getch()
                    ;
                    break;

                case 4:
                    b1.FindMin();
                    getch()
                    ;
                    break;

                case 5:
                    b1.CountNodes
                    (); getch();
                    break;

                case 6:
                    exit(1);

                default:
                    cout<<"Invalid
                    choice"; break;
            }
        }

}
```

Output:

```
C:\Users\Dell\Dropbox\DS\UNIT06\PRG_17.exe
*********Binary Search Tree(BST)*********

1.Insert Node in the binary search tree :
2.Display The BST :
3.Find the maximum node :
4.Find the minimum node :
5.Count the node in the BST :
6.Exit
Enter The choice : 1
Enter the number to Insert in BST: 43
```

```
C:\Users\Dell\Dropbox\DS\UNIT06\PRG_17.exe
*********Binary Search Tree(BST)*********

1.Insert Node in the binary search tree :
2.Display The BST :
3.Find the maximum node :
4.Find the minimum node :
5.Count the node in the BST :
6.Exit
Enter The choice : 1
Enter the number to Insert in BST: 23
Duplicate!!
```

```
C:\Users\Dell\Dropbox\DS\UNIT06\PRG_17.exe
*********Binary Search Tree(BST)*********

1.Insert Node in the binary search tree :
2.Display The BST :
3.Find the maximum node :
4.Find the minimum node :
5.Count the node in the BST :
6.Exit
Enter The choice : 2
Preorder : 23-43-32-65-end
Postorder : 32-65-43-23-end
Inorder : 23-32-43-65-end
```

C:\Users\Dell\Dropbox\DS\UNIT06\PRG_17.exe

```
*********Binary Search Tree(BST)*********

1.Insert Node in the binary search tree :
2.Display The BST :
3.Find the maximum node :
4.Find the minimum node :
5.Count the node in the BST :
6.Exit
Enter The choice : 3
Maximum Node is 65
```

C:\Users\Dell\Dropbox\DS\UNIT06\PRG_17.exe

```
*********Binary Search Tree(BST)*********

1.Insert Node in the binary search tree :
2.Display The BST :
3.Find the maximum node :
4.Find the minimum node :
5.Count the node in the BST :
6.Exit
Enter The choice : 4
Minimum Node is 23
```

C:\Users\Dell\Dropbox\DS\UNIT06\PRG_17.exe

```
*********Binary Search Tree(BST)*********

1.Insert Node in the binary search tree :
2.Display The BST :
3.Find the maximum node :
4.Find the minimum node :
5.Count the node in the BST :
6.Exit
Enter The choice : 5
Number Of the node in the BST : 4
```