# Program:

//Binary Search Tree-UR12CS135

#include<iostream>

using namespace std;

struct tree\_node

{

int data;

tree\_node \*left,\*right;

}\*root=NULL;

void inorder(tree\_node \*);

void postorder(tree\_node \*);

void preorder(tree\_node \*);

class Binary\_tree

{

private:

int r,c,choice;

public:

void creation()

{

cout<<"Enter the Root Node\n";

cin>>r;

if(root==NULL)

{

root=new tree\_node;

root->data=r;

root->right=NULL;

root->left=NULL;

}

}

void insertion()

{

cout<<"Insertion of node\n";

cout<<"Enter the Node\n";

cin>>c;

tree\_node \*newnode=new tree\_node;

newnode->data=c;

newnode->right=NULL;

newnode->left=NULL;

tree\_node \*temp;

temp=root;

tree\_node \*parent;

while(temp!=NULL)

{

parent=temp;

if(c>(temp->data))

temp=temp->right;

else

temp=temp->left;

}

if(c>parent->data)

parent->right=newnode;

else

parent->left=newnode;

}

void traversal()

{

cout<<"Select a Mode of traversal:\n\t1.In-order\n\t2.Post-Order\n\t3.Pre-Order\n";

cin>>choice;

switch(choice)

{

case 1:

inorder(root);

break;

case 2:

postorder(root);

break;

case 3:

preorder(root);

break;

}

}

};

void inorder(tree\_node \*temp1)

{

if(temp1!=NULL)

{

inorder(temp1->left);

cout<<temp1->data<<" ";

inorder(temp1->right);

}

else

return;

}

void postorder(tree\_node \*temp2)

{

if(temp2!=NULL)

{

postorder(temp2->left);

postorder(temp2->right);

cout<<temp2->data<<" ";

}

}

void preorder(tree\_node \*temp3)

{

if(temp3!=NULL)

{

cout<<temp3->data<<" ";

preorder(temp3->left);

preorder(temp3->right);

}

}

int main()

{

int ch,operation,repeat;

Binary\_tree B;

cout<<"\tUR12CS135- Binary Search Tree\n\n";

L5:

cout<<"\nIf You Want to create a root node Press 1 else 2\n";

cin>>ch;

if(ch==1)

goto L1;

if(ch==2)

goto L2;

L1:

B.creation();

cout<<"Select Your option\n";

L3:

cout<<"1.Insert\t2.Traverse\n";

cin>>operation;

switch(operation)

{

case 1:

B.insertion();

break;

case 2:

B.traversal();

break;

}

cout<<"If You Want To Repeat Press 1\n";

cin>>repeat;

if(repeat==1)

goto L3;

else

goto L4;

L2:

cout<<"No Existiong Nodes!!\n";

goto L5;

L4:

return 0;

}

# Output:

