# **Week-6**

**Implement a function that inserts a node in a given binary tree.**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package binarytree;

/\*\*

\*

\* @author Anonymous

\*/

public class BinaryTree {

class Node {

int key;

Node left, right;

public Node(int number) {

key = number;

left = right = null;

}

}

Node root;

BinaryTree() {

root = null;

}

void insert(int key) {

root = insertRecursive(root, key);

}

/\* A recursive function to insert a new key in BST \*/

Node insertRecursive(Node root, int key) {

/\* If the tree is empty, return a new node \*/

if (root == null) {

root = new Node(key);

return root;

}

/\* Otherwise, recur down the tree \*/

if (key < root.key)

root.left = insertRecursive(root.left, key);

else if (key > root.key)

root.right = insertRecursive(root.right, key);

/\* return the (unchanged) node pointer \*/

return root;

}

void inorderRec(Node root) {

if (root != null) {

inorderRec(root.left);

System.out.println(root.key);

inorderRec(root.right);

}

}

void inorder() {

inorderRec(root);

}

public static void main(String[] args) {

BinaryTree tree = new BinaryTree();

tree.insert(50);

tree.insert(30);

tree.insert(20);

tree.insert(40);

tree.insert(70);

tree.insert(60);

tree.insert(80);

// print inorder traversal of the BST

tree.inorder();

}

}