```
//Creating Binary Search Tree

class Node {
  Node left;
  Node right;
  int data;
}

class BST {

  public Node insert(Node node, int val) {
    if(node == null) {
      return createNewNode(val);
    }

    if(val < node.data) {
      node.left = insert(node.left, val);
    } else if((val > node.data)) {
      node.right = insert(node.right, val);
    }

    return node;
  }

  public Node createNewNode(int k) {
    Node a = new Node();
    a.data = k;
    a.left = null;
    a.right = null;
    return a;
  }
}

public class BSTApp {

  public static void main(String[] args) {
    BST a = new BST();
    Node root = null;

    root = a.insert(root, 8);
    root = a.insert(root, 3);
    root = a.insert(root, 6);
    root = a.insert(root, 10);
    root = a.insert(root, 4);
    root = a.insert(root, 7);
    root = a.insert(root, 1);
    root = a.insert(root, 14);
    root = a.insert(root, 13);

  }
}
```

BST - Print elements in Inorder (Sorted/ Ascending) in Binary Search Tree

```
class Node {
  Node left;
  Node right;
  int data;
}

class BST {

  public void inorder(Node node) {
    if(node == null) {
      return;
    }

    inorder(node.left);
    System.out.print(node.data + " ");
    inorder(node.right);
  }

  public Node insert(Node node, int val) {
    if(node == null) {
      return createNewNode(val);
    }

    if(val < node.data) {
      node.left = insert(node.left, val);
    } else if((val > node.data)) {
      node.right = insert(node.right, val);
    }

    return node;
  }

  public Node createNewNode(int k) {
    Node a = new Node();
    a.data = k;
    a.left = null;
    a.right = null;
    return a;
  }
}

public class BSTApp {

  public static void main(String[] args) {
    BST a = new BST();
    Node root = null;
```

```
    root = a.insert(root, 8);
    root = a.insert(root, 3);
    root = a.insert(root, 6);
    root = a.insert(root, 10);
    root = a.insert(root, 4);
    root = a.insert(root, 7);
    root = a.insert(root, 1);
    root = a.insert(root, 14);
    root = a.insert(root, 13);

    a.inorder(root);
  }

}
```