

Binary Search Trees

```
class BinaryTreeNode {
    int element;
    BinaryTreeNode left;
    BinaryTreeNode right;
    public BinaryTreeNode(int theElement) {
        element = theElement;
        left = null;
        right = null;
    }
}
```

Outline

```
class BinarySearchTree {  
    BinaryTreeNode root;  
    public BinarySearchTree(){  
    public boolean search(int theElement) {}  
    public void insert(int theElement){  
    public void printTree(){  
}
```

Constructor

```
public BinarySearchTree(){  
    root = null;  
}
```

Search

```
public boolean search(int theElement)
{
    BinaryTreeNode x = root;
    while (x != null) {
        if (x.element == theElement) return true;
        if (x.element < theElement) x = x.right;
        else x = x.left;
    }
    return false;
}
```

Insertion

```
public void insert(int theElement)
{
    if (root == null) {
        root = new BinaryTreeNode(theElement);
        return;
    }
    BinaryTreeNode x = root;
    boolean done = false;
```

Insertion (continued)

```
while (! done) {  
    if (x.element == theElement) return;  
    if (x.element < theElement) {  
        if (x.right == null) {  
            x.right = new BinaryTreeNode(theElement);  
            done = true;}  
        else { x = x.right; }  
    }  
    else {Code}  
}
```

```
if (x.left == null) {  
    x.left = new BinaryTreeNode(theElement);  
    done = true;}  
else { x = x.left;}
```


print

```
public void printTree(){  
    print(root);  
}
```

```
public void print(BinaryTreeNode x) {  
    if (x != null) {  
        print(x.left);  
        System.out.println(x.element);  
        print(x.right);  
    }  
}
```

Using the tree

```
class BST {  
    public static void main(String [] args) {  
        BinarySearchTree myTree = new BinarySearchTree();  
        int e;  
  
        for (int i = 0; i < 10; i++) {  
            e = (int) (20 * Math.random());  
            myTree.insert(e);  
            System.out.println(e);  
        }  
    }  
}
```

Using the Tree

```
for (int i = 0; i < 10; i ++ ) {  
    e = (int) (20 * Math.random());  
    System.out.println(e + " : " + myTree.search(e));  
}
```

```
myTree.printTree();  
}  
}
```