

# C++ LAUNCHPAD



## Lecture Data Structures

- Binary Search Trees

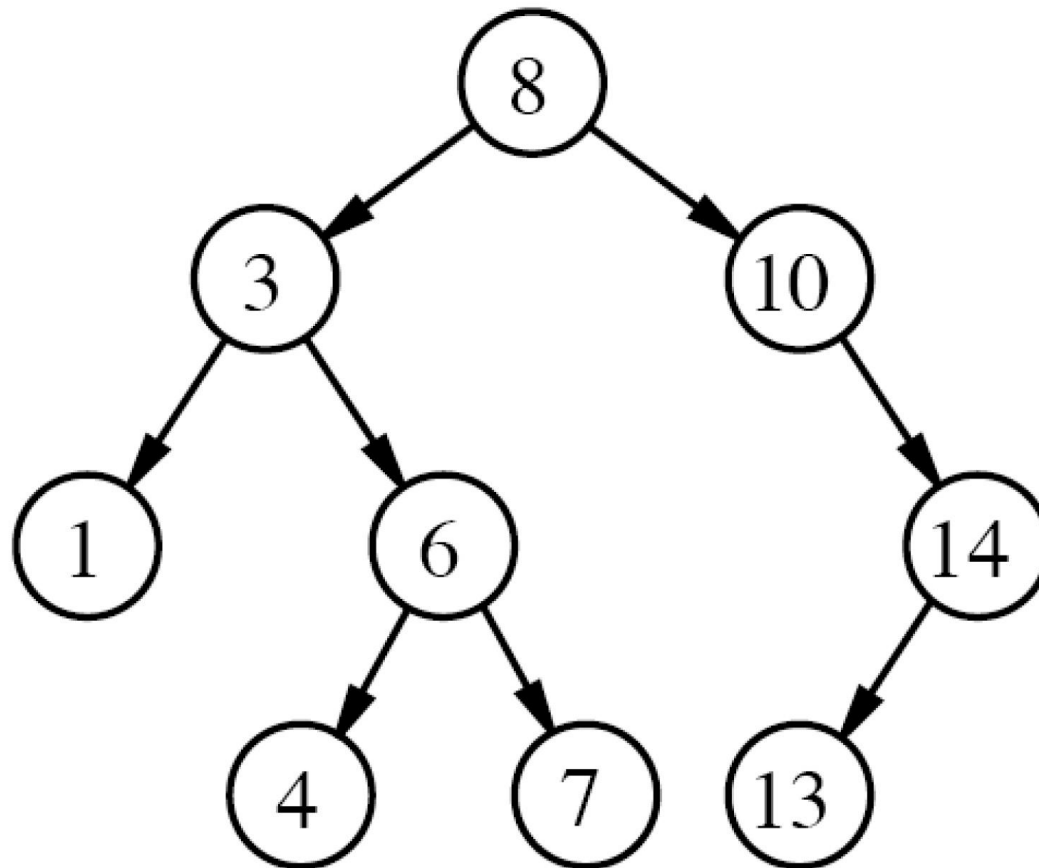
Deepak Aggarwal  
[deepak@codingblocks.com](mailto:deepak@codingblocks.com)

# Binary Search Trees

# BST Properties

1. Every Node in left subtree has value less than or equal to root
2. Every Node in right subtree has value greater than or equal to root

# Binary Search trees



# Binary Search Trees

```
class BinarySearchTree{  
    // accessor methods  
    int size();  
    bool isEmpty();  
    bool findElement(int element);  
    // update methods  
    void addElement(int element);  
    void removeElement(int element) throws  
        BSTEmptyException;  
}
```

# Lets discuss few problems

1. Print BST elements in range K1 and K2
2. Search & Adding element in BST

## Your Turn

1. Convert a BST into sorted Linked List
2. Given a binary tree check if its BST
3. Check if a Binary Tree is Balanced

# Build a BST using a sorted array



# Balanced/unbalanced Tree

# Balanced Trees

1. AVL Tree
2. Red Black Trees
3. 2-4 Trees

# C++ LAUNCHPAD



Thank You!

Deepak Aggarwal