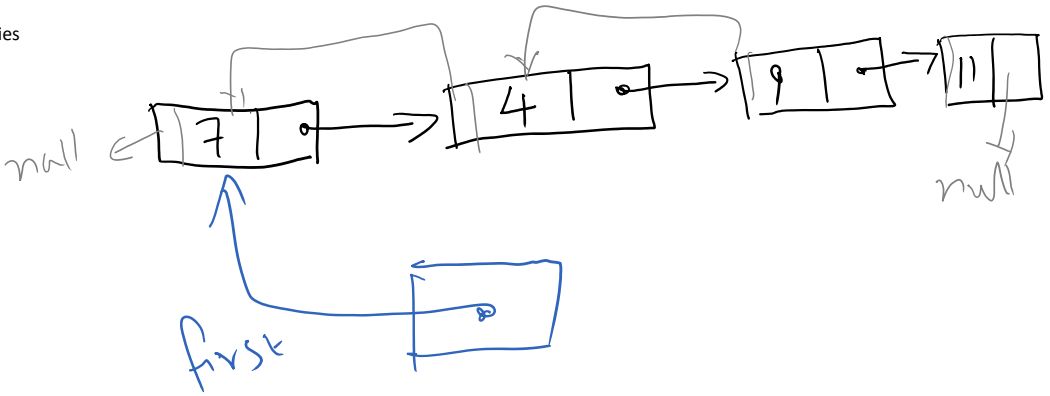
Assignment01 LinkedList

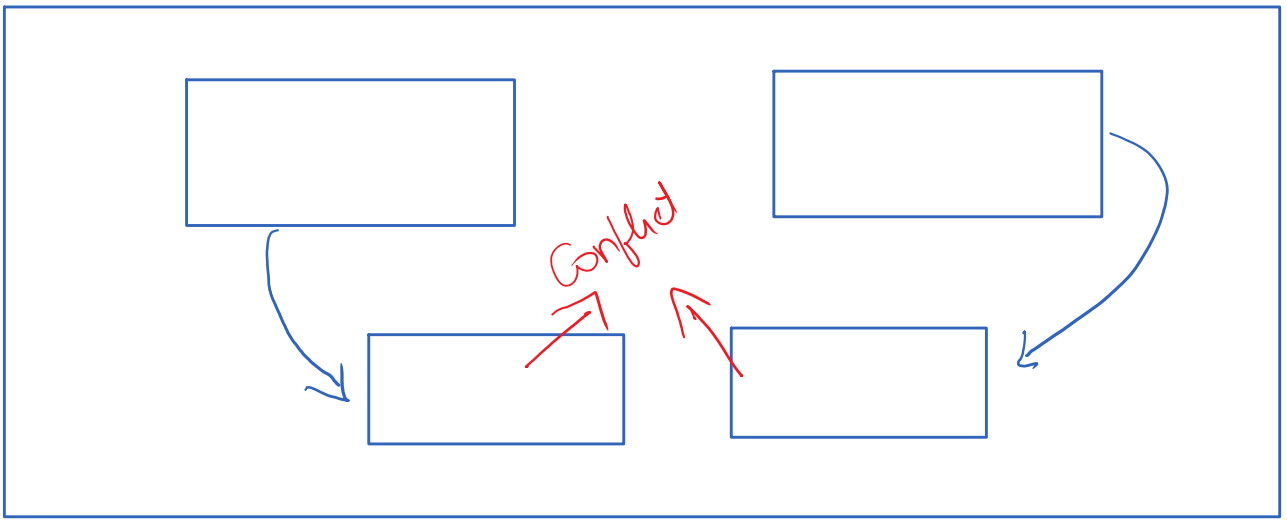
Thursday, May 21, 2020 2:58 PM

- Create a class to represent a Linked List
- A Linked List should support following operations
  - **add(int value)** //Adds to end of the List
  - **get(int pos)** //get a value from a given position
  - **set(int pos)** //set a valjue to a given position
  - **size()** //returns the size of the list
  - **remove(int pos)** //remove the value from a given position
- Create the necessary classes
- Write a **main function** to test its functionalities

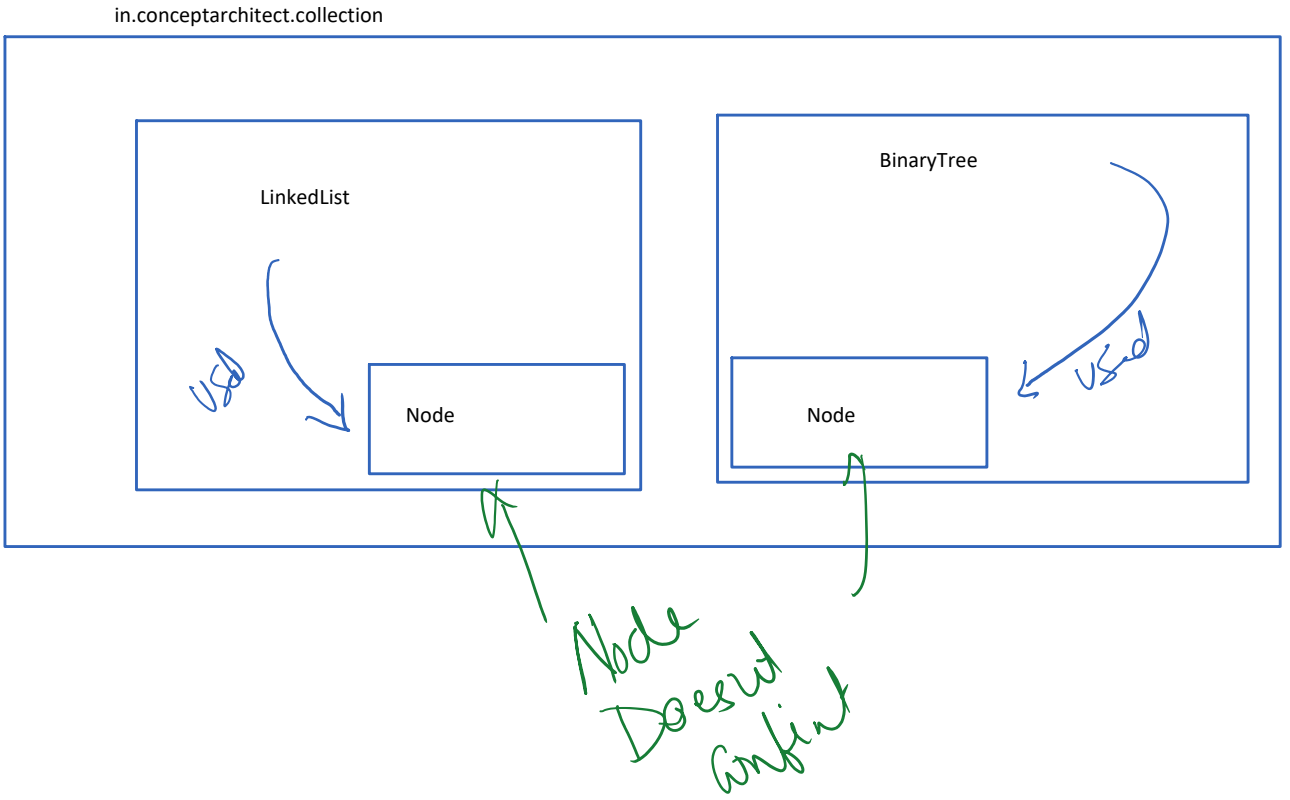


Package vs Class Boundry

Friday, May 22, 2020 3:57 PM



A class to can act as a Package to separate class name visibility



When should I user inner class

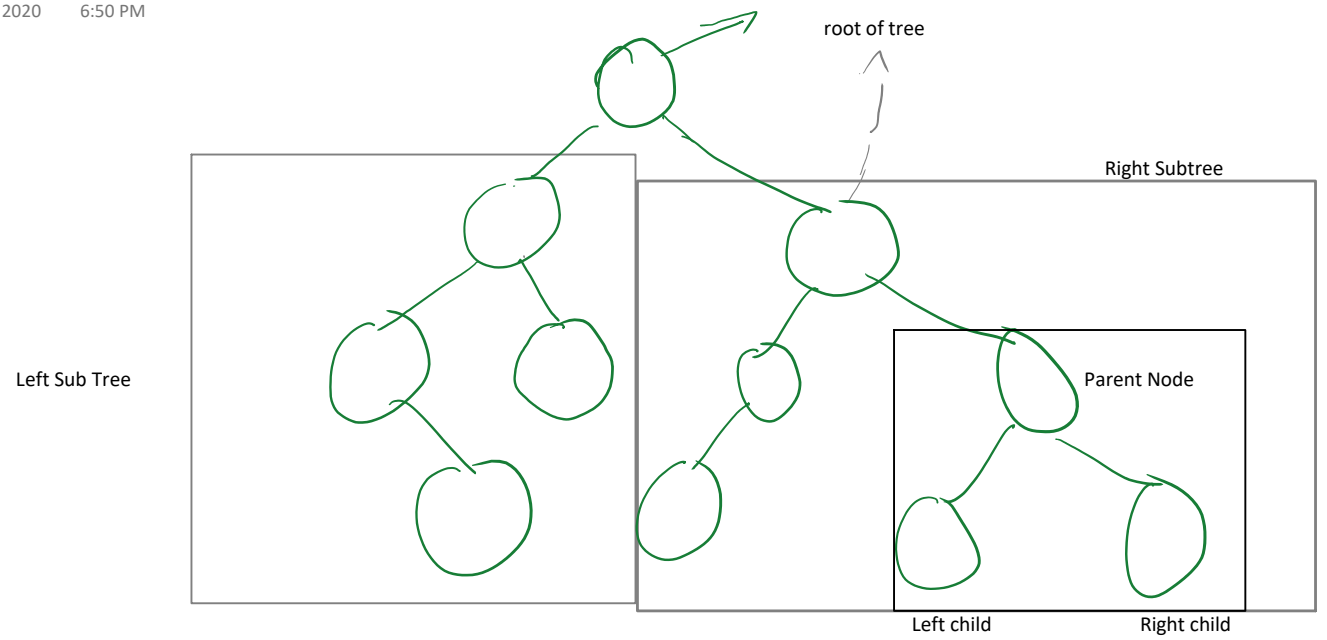
- The outer class uses the objects of inner class **exclusively**
- The inner class object is not directly utilized by anyone else
- The only purpose of inner class is to support the outer class

Not every child component should be inner class

- A car contains tyres
- But a Tyre has independent existence and manufacturer
- We will not define Tyre class as inner class to Car

BinaryTree of int

Friday, May 22, 2020 6:50 PM



BinaryTree Create Rule

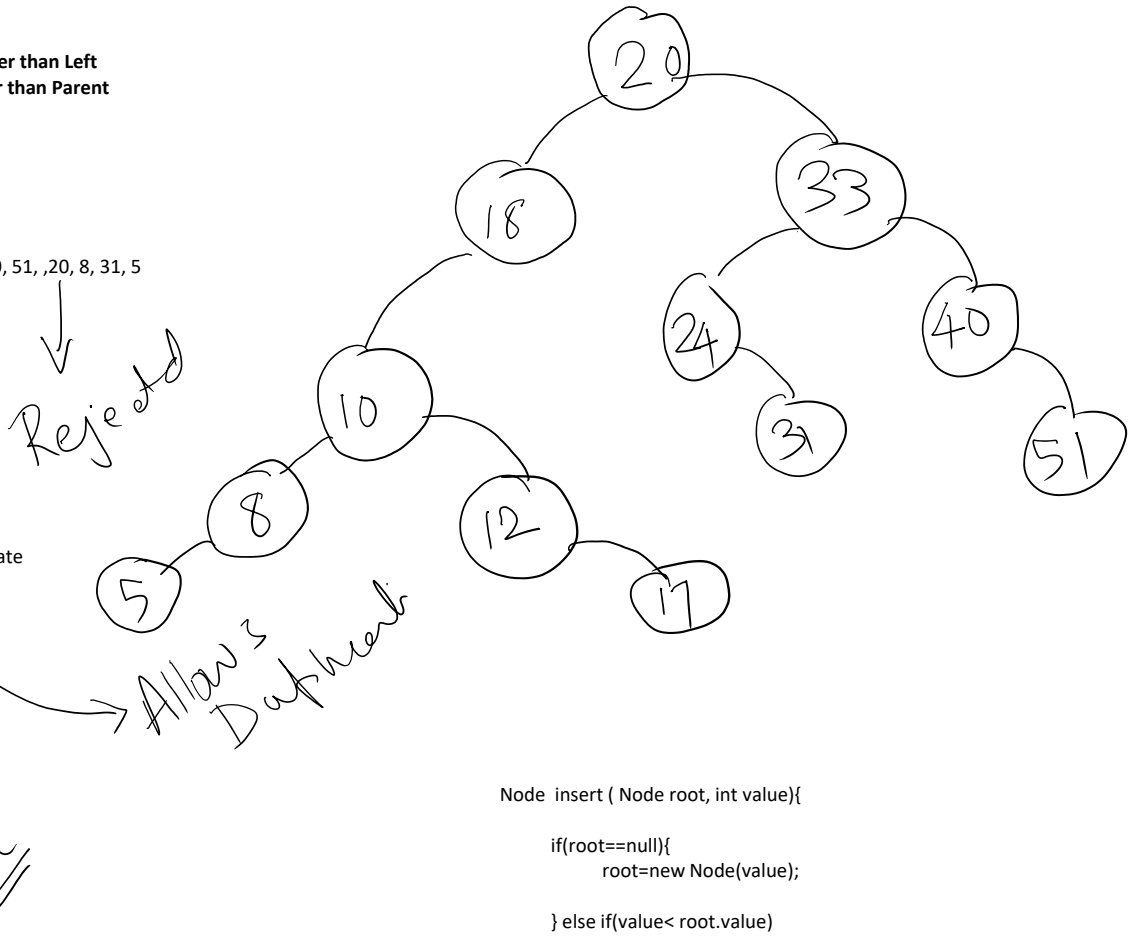
Friday, May 22, 2020

Tree Rule: L<P<R

- **Parent should be greater than Left**
- **Right should be greater than Parent**

To Add following Numbers

20 , 18, 10, 12, 17, 33, 24, 40, 51, ,20, 8, 31, 5



- If you want to retain duplicate you can change the tree forumula to one of the two given below
  - L<=P < R
  - or L<P<=R
- But Not
  - L<=P<=R

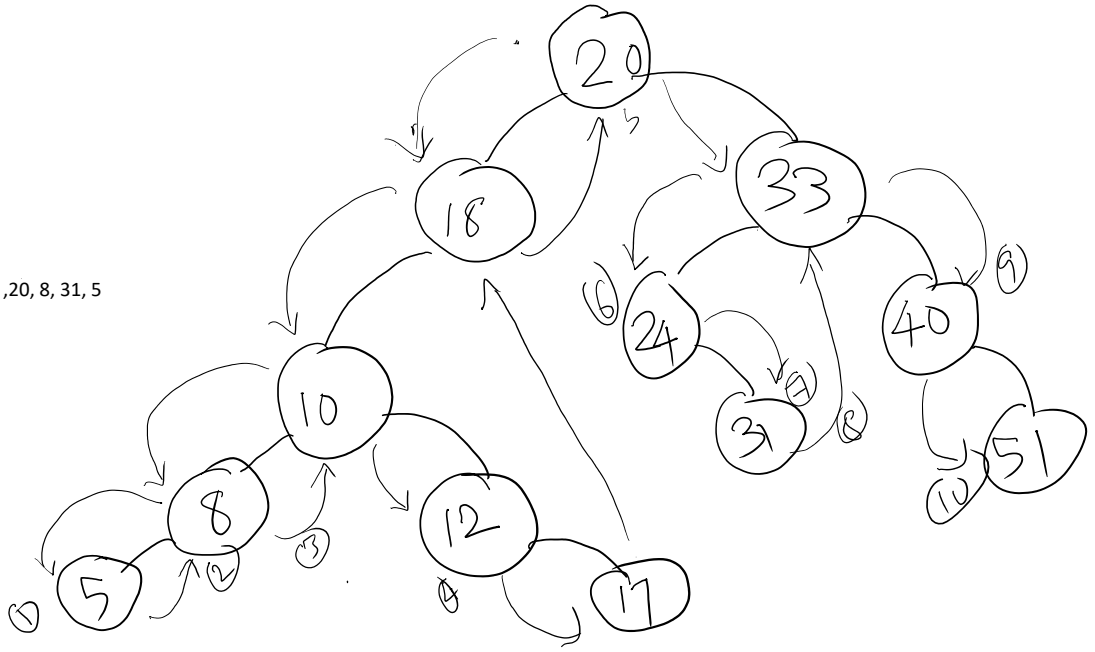

```
Node  insert ( Node root, int value){

        if(root==null){
                root=new Node(value);

        } else if(value< root.value)
                root.left=insert(root.left,value);
        else if(vlaue> root.value)
                root.right=insert(root.right,value);
        return root;
}


}
```

BinaryTreeAccess Rule -- Inorder

Friday, May 22, 2020

Inorder: L-->P-->R

- **Visit Left (subtree)**
- **Visit Parent**
- **Visit Right(subtree)**



To Add following Numbers

20 , 18, 10, 12, 17, 33, 24, 40, 51, ,20, 8, 31, 5

Inorder

5
8
10
12
17
18
20
24
31
33
40
51

Preorder

P --> L --> R

Preorder

L --> R --> P

```
Node inorder ( Node root){

        if(root==null){
                return;

        }
        inorder(root.left); //L
        print(root.value);  //P
        inorder(root.right); //R


}
```

Assignment 02

Friday, May 22, 2020 7:10 PM

- create class BinaryTree to store integers
- Implement operations
    - Insert
    - Inorder
    - Preorder
    - Postorder