Class Header

```
public class Tree1 {
  private int data;
  private Tree1 left,right;
  private boolean empty;

  public Tree1() {
    empty=true;
  }
```

What is `empty` doing?

Constructor that Takes Data

```
public Tree1(int d) {
  empty=false;
  data=d;
  left=new Tree1();
  right=new Tree1();
}
```


Constructor that takes Data and more trees

```
public Tree1(int d, Tree1 l, Tree1 r) {
  empty=false;
  data=d;
  left=l;
  right=r;
}
```


Getting Info

```
public boolean isEmpty() {
  return empty;
}

public int info() {
  return data;
}
```

Getting Subtrees

```
public Tree1 left() {
  return left;
}

public Tree1 right() {
  return right;
}
```


Some Algorithms

```
public int number() {
  if (isEmpty()) return 0;
  else return 1 + left().number() + right.number();
}
```



```
public void printPre() {
  if (isEmpty())
    return;
  System.out.println(info());
  left().printPre();
  right().printPre();
}
```



```
public void printIn() {
  if (isEmpty())
    return;
  left().printIn();
  System.out.println(info());
  right().printIn();
}
```

```
public boolean isPerfectBal(){
  if (isEmpty()) return true;
  if (((abs(left().number() - right().number())) <= 1) &&
      left().isPerfectBal() && right().isPerfectBal()) return true
  else return false;
}
```



```
public void printPost() {
  if (isEmpty())
    return;
  left().printPost();
  right().printPost();
  System.out.println(info());
}
```


Test Code

```
public static void main(String[] arg) {
  Tree1 t1=new Tree1(1);
  Tree1 t2=new Tree1(2);
  Tree1 t3=new Tree1(3,t1,t2);
  //    Node1 n= t3.pre();
  // n.print();
  //System.out.println("Sum"+t3.sum());
  System.out.println("Number"+t3.number());
  System.out.println("isPerfBal "+t3.isPerfectBal());
  t3.printPre();
  t3.printIn();
  t3.printPost();
}
```