Question 1

```
import java.util.*;

class Clock {

    int hr, min;

    Clock(int hr, int min){
        this.hr = hr;
        this.min = min;
    }

    public double calculateHourHand(){
        //Considering 12 as origin. When hour hand between two, using min/60 we can calculate where exactly is the hr hand in between.
        //For 12 hr 360 degree, so for 1 hr 30 degree

        double h_angle = (hr + min/60.0) * 30;
        return h_angle;
    }

    public int calculateMinuteHand(){

        //For 60 min 360 degree, so for 1 min 6 degree

        int m_angle = min * 6;
        return m_angle;
    }

    public double findAngle(){

        double diff;
        double h_angle = calculateHourHand();
        int m_angle = calculateMinuteHand();

        if(m_angle>h_angle) diff = m_angle - h_angle;
        else                diff = h_angle - m_angle;

        if(diff > 180)  return (360-diff);
        return diff;
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int hr, min;

        System.out.print("Enter time hour : ");
        hr = sc.nextInt();
        System.out.print("\nEnter time minute : ");
        min = sc.nextInt();

        Clock C1 = new Clock(hr,min);
        System.out.println("\nAngle between hour and minute hand is : " + C1.findAngle());
    }
}
```

QUESTION 2

```
import java.util.*;

class Branch {

    List<Branch> branches;
    int val;

    Branch(){
        val = 0;
        branches = new ArrayList<>();
    }

    Branch(int val){
        this.val = val;
        branches = new ArrayList<>();
    }

    public void add(Branch b){
        branches.add(b);
    }

    public int findDepth(){

        if(branches.size() == 0)    return 1;

        int max = Integer.MIN_VALUE;

        for(int i=0 ; i<branches.size() ; ++i){
            Branch cur = branches.get(i);
            int dep = cur.findDepth();
            max = Math.max(dep,max);
        }

        return 1+max;
    }

    public static void main(String[] args) {

        //lets make a structure like this :
        //                   1
        //               2       3
        //             4   5   6  7
        //           8   9
        //        10
        //Inorder traversal : 1,2,4,8,10,9,5,3,6,7
        //depth = 5

        Branch b1 = new Branch(1);
        Branch b2 = new Branch(2);
        Branch b3 = new Branch(3);
        Branch b4 = new Branch(4);
        Branch b5 = new Branch(5);
        Branch b6 = new Branch(6);
        Branch b7 = new Branch(7);
        Branch b8 = new Branch(8);
        Branch b9 = new Branch(9);
        Branch b10 = new Branch(10);

        b8.add(b10);

        b4.add(b8);
        b4.add(b9);

        b2.add(b4);
        b2.add(b5);

        b3.add(b6);
        b3.add(b7);

        b1.add(b2);
        b1.add(b3);

        System.out.println("Depth : " + b1.findDepth());
    }
}
```