Day 17

Nested Class

- We can define class inside scope of another class. It is called as nested class.
- Consider Example:

```
class Node{
    //TODO : Write code here
}
class LinkedList{
    private Node head;
    private Node tail;
    public void addLast( int element ){
        Node newNode = new Node( element );
        //TODO : Write code here
    }
    public void print( ){
        Node trav = this.head;
        //TODO : Write code here
    }
}
```

```
class LinkedList{   //Top Level Class
    class Node{ //Nested class
        //TODO : Write code here
    }
    private Node head;
    private Node tail;
    public void addLast( int element ){
        Node newNode = new Node( element );
        //TODO : Write code here
    }
    public void print( ){
        Node trav = this.head;
        //TODO : Write code here
    }
}
```

- By defining nested class, we implement encapsulation.

```
//Top Level class
class Outer{    //Outer.class
    //Nested class
    class Inner{    //Outer$Inner.class
```

```
    }
}
```

- Access modifier of top level class can be package level private or public only.
- Access modifier of nested class can be private, package level private, protected or public.

Types of nested class:

```
1. Non static nested class / Inner class
2. Static nested class
```

```
class LinkedList implements Iterable<Integer>{
    //Static nested class
    class Node{
        int data;
        Node next;
    }

    private Node head;
    private Node tail;
    public Iterator<Integer> iterator( ){
        Iterator<Integer> itr = new LinkedListIterator( this.head );
        return itr;
    }
    //Non static nested class
    class LinkedListIterator implements Iterator<Integer>{
        private Node trav;
        public LinkedListIterator( Node head ){
            this.trav = head;
        }
    }
}
```

Non Static Nested class.

- Non Static Nested class is also called as inner class.
- If implementation of nested class depends on top level class then nested class should be non static.
- In above code, LinkedListIterator depends on LinkedListIterator hence it should be non static.

```
//Top Level class
class Outer{    //Outer.class
    //Nested class
    public class Inner{    //Outer$Inner.class

    }
}
```

- Hint : For simplicity, consider non static nested class as a non static method of a class.
- Instantiation of top level class:

```
Outer out = new Outer( );
```

- Instantiation of non static nested class:
- First way

```
Outer out = new Outer( );
Outer.Inner in = out.new Inner( );
```

- Second way

```
Outer.Inner in = new Outer( ).new Inner( );
```

Static Nested class.

- When we declared nested class static then it is simply called as static nested class.
- If implementation of nested class do not depends on top level class then nested class should be static.
- In above code, Node do not depends on LinkedListIterator hence it should be static.

```
//Top Level class
class Outer{    //Outer.class
    //Nested class
    public static class Inner{    //Outer$Inner.class

    }
}
```

- We can not declare top level class static but we can declare nested class static.
- Hint : For simplicity, consider static nested class as a static method of a class.
- Instantiation of top level class:

```
Outer out = new Outer( );
```

- Instantiation of static nested class:

```
Outer.Inner in = new Outer.Inner( );
```