Checking Balance

```
class StackObject {
    char cr;
    int    loc;
    public StackObject(char c, int l) {
        cr = c;
        loc = l;
    }
}
```

The Stack

```
class MyStack {
    StackObject [] arr;
    int StackTop;
    public MyStack() {  code
    }

    public void push (StackObject o) { code
     }
    public StackObject pop () { code
    }
}
```

```
public MyStack() {
    arr = new StackObject[20];
    StackTop = -1;
}

public void push (StackObject o) {
    StackTop++;
    arr[StackTop] = o;
}
```

```
public StackObject pop () {
if (StackTop == -1) {return null;}
    else {
        StackTop--;
        return arr[StackTop + 1];
    }
}
```

```
class Auxiliaries {
    public static boolean matches(char a, char b) {
        boolean ans = false;
        if (( a == '(') && (b == ')' )) ans = true;
        if ((a == '[' ) && (b == ']' )) ans = true;
        return ans;
    }
}
```

```
public class Paranth {
    public static void main (String argv []) {
    code
    }
}
```

Code in main method

```
char [] str = {'(', '[', ']', '[', ']',']'};
StackObject ObjectOnTop;
int i =  0;
boolean notDone = true;
MyStack sta = new MyStack();
```

main: the while loop

```
while( (i < str.length) && notDone) {

   if ( (str[i] == '(') || (str[i] == '[')) {
   code1
   }
   else if( (str[i] == ')') || (str[i] == ']') ) {
   code2
   }
   i++;
}
```

code1

```
sta.push(new StackObject(str[i],i));
```

code2

```
ObjectOnTop = sta.pop();
 // pop otherwise, and process as below
 if (ObjectOnTop == null) {
    notDone = false;
    Print error message
 }
 else if (! Auxiliaries.matches(ObjectOnTop.cr, str[i]) ) {
    notDone = false;
    Print error message
 }
```

main: Outside the while loop

```
if ( notDone && (sta.pop() != null)  )
        System.out.println("Paranths need to be closed");
```