

# Homework Turnin

Account: 6G\_06 (rgalanos@fcps.edu)  
Section: 6G  
Course: TJHSST APCS 2016-17  
Assignment: 06-01  
Receipt ID: 301f81fca00b3274ecc793467b6ec508

## Turnin Successful!

The following file(s) were received:

### ParenMatch.java (1757 bytes)

```
1. // name:    date:
2.
3. import java.util.*;
4. public class ParenMatch
5. {
6.     public static final String left = "([{<";
7.     public static final String right = ")]}>";
8.     public static void main(String[] args)
9.     {
10.        System.out.println("Parentheses Match");
11.        ArrayList<String> parenExp = new ArrayList<String>();
12.        /* enter tests here */
13.        parenExp.add("5+7");
14.        parenExp.add("(5+7)");
15.        parenExp.add(")5+7(");
16.        parenExp.add("((5+7)*3)");
17.        parenExp.add("<{5+7}*3>");
18.        parenExp.add("[ (5+7)*13");
19.        parenExp.add("(5+7)*3");
20.        parenExp.add("5+(7*3)");
21.        parenExp.add("((5+7)*3)");
22.        parenExp.add("[ (5+7)*3)");
23.        parenExp.add("[ (5+7)*3]");
24.        parenExp.add("[(5+7)*3]");
25.
26.        for( String s : parenExp )
27.        {
28.            boolean good = checkParen(s);
29.            if(good)
30.                System.out.println(s + "\t good!");
31.            else
32.                System.out.println(s + "\t BAD");
33.        }
34.    }
35.    public static boolean checkParen(String exp)
36.    {
37.        Stack<String> stack = new Stack<String>();
38.
39.        for(int x=0;x<exp.length();x++)
40.        {
41.            if(left.indexOf(exp.charAt(x))!=-1)
42.                stack.push(""+exp.charAt(x));
43.            else if(right.indexOf(exp.charAt(x))!=-1)
44.            {
45.                if(stack.isEmpty())
46.                    return false;
47.                if(left.indexOf(stack.pop())!=right.indexOf(exp.charAt(x)))
48.                    return false;
49.            }
50.        }
```

```
51.     if(stack.isEmpty())
52.         return true;
53.     else
54.         return false;
55. }
56. }
57.
58. /*
59.  Parentheses Match
60.  5+7    good!
61.  (5+7)  good!
62.  )5+7(  BAD
63.  ((5+7)*3) good!
64.  <{5+7}*3> good!
65.  [(5+7)*3] good!
66.  (5+7)*3 good!
67.  5+(7*3) good!
68.  ((5+7)*3 BAD
69.  [(5+7)*3] BAD
70.  [(5+7)*3]) BAD
71.  ([ (5+7)*3] BAD
72.  */
73.
```