

Balanced Brackets

Create a new Java project **stackBrackets**.

Create classes **BalancedBrackets** and **BalancedBracketsTester**.

Coding interviews might ask you to write a program that will check a string for balanced brackets: each **opening** bracket, brace, or parentheses should have a matching **closing** one.

One way to do this is to use a stack:

- loop through a string
- if a character is an opening bracket
 - push it onto the stack
- if a character is a closing bracket
 - pop the top of the stack
- when the loop is finished
 - if there is an even number of opening and closing brackets
 - the stack will be empty and brackets are **balanced**
 - else there are unmatched brackets and they are **unbalanced**

Add the code for this in **BalancedBrackets**:

```
1 package submissions;
2
3 import java.util.Stack;
4
5 public class BalancedBrackets
6 {
7     public String checkBalance(String string)
8     {
9         Stack<Character> stack = new Stack<Character>();
10
11         for (int i = 0; i < string.length(); i++)
12         {
13             char c = string.charAt(i);
14             if (c == '(')
15             {
16                 stack.push(c);
17             }
18             else
19             {
20                 stack.pop();
21             }
22         }
23
24         if (stack.empty())
25         {
26             return "balanced";
27         }
28         else
29         {
30             return "unbalanced";
31         }
32     }
33 }
```

Test the code in **BalancedBracketsTester**:

```
1 package submissions;
2
3 public class BalancedBracketsTester
4 {
5     public static void main(String[] args)
6     {
7         BalancedBrackets balanced = new BalancedBrackets();
8
9         System.out.println(balanced.checkBalance("()"));
10        System.out.println(balanced.checkBalance("(()())"));
11        System.out.println(balanced.checkBalance("((((()()()))));
12        System.out.println(balanced.checkBalance("()"));
13    }
14 }
```

This will work with balanced brackets and unmatched opening brackets. However, if you have unmatched **closing** brackets it will not work.

Fix the code so that it will be able to return *unbalanced* for unmatched **closing** brackets also. For example, “`)`” or “`)()`”.

Optional extra credit points: 10

Create a class **BalancedAllBrackets** and modify the original code so that it will check the balance of not just parentheses, but curly braces and square brackets also: “([{ }])”, “{ } () }”.

Zip and submit the **project** folder.

Leave a comment when you submit if you attempt the extra credit activity.