

CMPS 2143 Object-Oriented Programming
Programming Assignment 1 50 points
DUE: Wednesday, Sept 13, 2017 @ 1pm

Purpose: To implement a dynamic array-based stack and use it to solve a problem; to implement interactive I/O and file output. **Program can be done in Java or C++.**

Problem: Palindromes - Write a program that uses a stack to determine whether a string entered at the keyboard has balanced parentheses. A string of characters has balanced parentheses if each right parenthesis occurring in the string can be matched with a preceding left parenthesis.

Input: ~~All input is from the keyboard and is of type string.~~ **Run your program on the following sample input. Also, make up your own data for a second run.**

Sample input for one run (not including the prompts):

```
()  
(()  
(())  
(I think(therefore I am))  
(())()()  
((()))  
(((())  
(as((b)c))  
(((())()  
(((())())
```

Output: *All output goes to both the screen and to a file. Format your output.* As usual, you may be creative with your output design.

C++ vs Java Requirements:

C++

- Must have .h, .cpp files
- Must have output file and display
- Read character by character from input file

Java

- No interface file required
- Only output to display
- Read whole string From keyboard

C++ Hints:

1. Read `infile.get(ch);` to get one character from a line
2. Use `+` to concat a string with char or string, e.g. `line = line + ch;`
3. Re-initialize `line=""`; the empty string before the while not `'\n'` loop

Java Hints:

- 1) Use `ch = s.charAt(i)` to get a character from a string
- 2) Use `+` to concat a string with other types (built-in types will convert to string)
- 3) Use `s.length()` to get the number of characters in a string

- Turn in:**
- source code listings (class (1-2), main)
 - 2 input files
 - 2 output screens and/or files for two runs of the program
 - complete project folder on some storage media
 - in a 9x12 envelope