Due: 7/26 (11:59PM)

STL Stack

- Write a C++ program `hw6.cpp` that does the following:
  - The program first creates an STL `stack` that stores `char` data.
  - Then ask user to enter a series of parentheses and/or braces, then report whether or not they're properly nested:

    Example run:

    ```
    Enter parentheses and/or braces: (){}{()})
    Parentheses/braces are nested properly
    Continue? (y/n)

    Enter parentheses and/or braces: {{{()}}
    Parentheses/braces are NOT nested properly
    Continue? (y/n)

    Enter parentheses and/or braces: {({{()(){}}))}
    Parentheses/braces are NOT nested properly
    Continue? (y/n)
    ```

  - After each round, ask user if he/she wants to continue. If yes, go back to the previous step and take another string of parentheses/braces. If no, terminate program.
  - *Hint*: As the program reads characters, have it push each left parenthesis or left brace into the STL stack. When it reads a right parenthesis or brace, have it pop the stack and check if the popped item is a matching parenthesis or brace (If not, the parentheses/braces aren't nested properly). When there is no more character to read, check if the stack is empty; if so, the parentheses/braces are matched.
  - If the user enters any character other than parenthesis or brace, simply ignore it. Therefore, entering a wrong character shouldn't lead to program malfunction.
  - Note: If you implement it *without* using STL stack, there will be a huge loss of points (even if the program works).

What to submit:

- All source files (.cpp, .h) needed for compilation (do not submit the entire Project)

How to submit:

- Use Canvas Assignment Submission menu to submit the assignment electronically at Canvas.
- Make sure to zip all your files into `hw6.zip`, then submit your `hw6.zip` as a single file.

Policy

- Make sure all your C++ programs properly compile and run on Eclipse C++.
- Projects will be graded 20% on style/standards and 80% on proper execution. Make sure to follow the *Coding Standards* posted on the course webpage. In particular, use proper indentation on each line of your code.
- At the beginning of each file (.cpp, .h), provide comments specifying the author, date, and a brief description of the file.
- Each source file (.cpp, .h) must contain enough comments here and there to make it easy to follow your code. Insufficient comments could lead to loss of points.
- Non-compilable program will get almost no credit (e.g., executable code not produced due to compile errors).
- Non-working program will get almost no credit (e.g., the executable is terminated immaturely due to run-time errors).
- Copying other's code is strictly prohibited. If identical (or nearly identical) submissions are found among students, every student involved will get automatic zero for the assignment. The same goes for copying existing source code from the internet.