LW: Working with Classes - String Class

# Objectives

* Use a built in C++ class.
* Get experience with message passing (i.e. using methods to communicate).

# Completion

* Work with others in your group, but submit individually.
* Each problem is worth 20 points.
* **You need 40 points on Gradescope for completion.**
  + **Pick any two of five problems to complete.**
  + After the lab, we suggest completing all of the problems to help study for the exam.

***Submission***

Submit the following files to the autograder.

* functions.cpp

#### ***Allowed Includes***

* iostream
* string
* stdexcept
* limits
* tuple
* sstream
* functions.h

# What To Do

#### ***Starter Code***

[Get starter code](https://drive.google.com/drive/folders/1NbRrKu-FDTxdwE6re_RL_1s1XuavdhfT?usp=sharing).

#### ***Recommendations***

* The focus should be on using the string class.
  + Communicate with the string objects using the methods/functions in the class.
* Pick your two favorite problems.
  + You should eventually do all five for practice.
* Use the string.at(int) function rather than string[] since this is the more object oriented approach.
  + Plus it throws exceptions when you use bad indexes which helps you to debug faster.’
* [zyBook sections 3.1 to 3.4](https://learn.zybooks.com/) are good references.
* Look at [string documentation](https://cplusplus.com/reference/string/string/) to find useful functions.
* A driver program is provided to run each of the functions.
  + The functions will not return or accept anything. Instead, they will use cout and cin to interact with I/O directly.
  + You will only need to make changes to functions.cpp
* Problem options:
  + [Sentence Deobfuscate](#_1t3h5sf)
  + [Word Filter](#_2s8eyo1)
  + [(In)Secure Password Converter](#_3rdcrjn)
  + [Number Word Calculator](#_lnxbz9)
  + [Palindrome Counter](#_1ksv4uv)

## 1. Sentence Deobfuscate

* Prompt the user to enter a collection of sentence words (i.e., words in the sentence), with the spaces removed (i.e., the obfuscated sentence) and with words that are less than ten (10) letters each.
* Prompt the user to enter a sequence of numbers that represent the length of each corresponding sentence word (i.e. the deobfuscated details).
* Output the deobfuscated sentence.
* Hints
  + Don’t overthink this problem.
  + Convert char c to int by subtracting 48 (‘0’) from c .

### Example

| Please enter obfuscated sentence: Thisisasentence  Please enter deobfuscation details: 4218  Deobfuscated sentence: This is a sentence |
| --- |

* This (4 letters), is (2 letters), a (1 letter), sentence (8 letters), STOP

## 2. Word Filter

* Prompt the user to enter a sentence and a filter word.
* Output the updated sentence with the filter word replaced with number signs (#).

### Example

| Please enter the sentence: One fish, two fish, red fish, blue fish.  Please enter the filter word: fish  Filtered sentence: One ####, two ####, red ####, blue ####. |
| --- |

## 3. (In)Secure Password Converter

* Prompt the user to enter text.
* Replace a subset of letters in the text with corresponding symbols as shown in the table below to create the new password.
* Append the reversed version of the new password to the new password.
* Output the original password and the new password.

| **Replace:** | **With:** |
| --- | --- |
| 'a' | '@' |
| 'e' | '3' |
| 'i' | '!' |
| 'o' | '0' |
| 'u' | '^' |

### Example

| Please enter your text input: password  input: password  output: p@ssw0rddr0wss@p |
| --- |

## 4. Number Word Calculator

* Prompt the user to enter a sequence of numbers and arithmetic symbols as words.
* Calculate the result of the equation.
* Output the words converted to numbers and symbols and the result.
* Valid number words: zero, one, two, three, four, five, six, seven, eight, nine
* Valid symbol words: plus, minus, times, divides
* Order of operation is simply left to right. That is, calculate immediately after reading two numbers and one arithmetic symbol.
* Use the std::to\_string function to convert an int to a string. E.g.,
  + string numberText = std::to\_string(9); // string numberText = "9";

### Example

| Please enter word equation (type 'equals' at the end):  five times two plus four divides seven equals  5 \* 2 + 4 / 7 = 2 |
| --- |

## 5. Palindrome Counter

* A palindrome is a word that looks the same when spelled forward and backward.
* Your code must prompt the user to enter a sequence of words, identify which words are palindromes, and output the number of palindromes.
* Enter the word 'quit' to indicate the end of the words to check.
* At the end, display the number of palindromes that the user had typed as shown in the sample execution
  + **note:** make sure that the result sentence is grammatically correct.

### Example

| Please enter a sentence (end with 'quit'): my favorite car is a racecar quit  You have typed 2 palindromes. |
| --- |

* Any string containing just one letter is by default a palindrome