Structure of Programming Languages Project

By David Brooks : A demonstration of two multiple paradigm languages (C++ and Python 3.7)

### What does the program do? Ask for a comma spepterated list of words and determines which ones supplied are palindromes or not.

## How to run it? The source code used for running the program is located in the **Exacutables directory**. ### C++ Verson: Since the compiled program is operating system dependent, please download the one that corresponds to your system `exe := Windows-x64` and `bin := Unix`. Depending on your security preferences, you may have to allow the program to run as it is not signed by any centural authority.

### Python 3 version: Due to the nature of Python and how it uses a interperator to function, you will need to intall the latest version of Python for your operating system. All the modules are zipped in the `Python.7z` file. This file is a zip file that requires the 7zip application, which can be installed here. After download simply open a termanal or command line at the directory location or unzipped the file to. Then emter `python ./Main_Python.py`

### How does it do it? Both versions will ask the user for input, store the input into a string and convert it into a vector or list container. This is then used to create a defined palindrome object. Each object has its own methods that determines if each word in the passed container is evaluated and used to determine its palindrome status. If it is a palindrome, it will be appeneded to a container that stores all positive palindromes.

## What did I learn about the differences between the two languages? As stated in the comments in the Python version, I am new to that language as most of my college programming coursework has been programmed in C++. Both of these languages are considered to be multi-paradigm languges. In my project I emphasized the OOP or object-oriented paradigm but used multiple paradigms in my program.

Comparing the languages

The biggest difference between the two languages that I noticed was that even though C++ is considered to be a high-level language when compared to all other languages, it appears to be quite the opposite when comparing it to Python. Python I found has a lot of high-level built in functions that in C++ you have to implement yourself or include from a third-source API.

For example the `.split()` function referenced in **Project1_DBrooks_Python.p1 -> `welcome_init()` function** alows me to simply take the users input string and return a list container that is deliminated by the "," seperator.

Where in C++ when the object was created the constructor calls `Palindrome::InitiateList()` I used a string stream to allow me to evaluate the character one char at a time and add the created word to the container when I was at the end of the string of provided words or when the null terminator char was encountered.

Conclusion:

Everthing in the world is never perfect and there is alsways some downside and a upside to something, and programming languages are no different. I think thats why Python is often used with other languages like C++. While having nice highlevel included methods are nice, there are times when a programmer wants to specifically spell something out persay. For example there is no easy way to do forward class or function declarations in Python, which can be a pain if you are used classes and implementing functions together. Thats why there is no one standard language for a specific area or problem, the best is to implement them together so we can "cherry pick" per say.