Objectives	Grow with Google Test Answer, b.hogan@snhu.edu

Program / operating parameters:

1. Demonstrate how to create a small Python program, called a script, and generate speech to text and text to audio results.
2. Challenge a user to replicate proper syntax, indenting, and other coding idioms to ensure programs run as intended.
3. Educate on basic data encoding where binary (1 or 0) is used for pictures/voice and nonbinary (byte/collations) is for text.
4. Educate on how libraries simplify program feature engineering making the art of the possible a far less daunting task.

Scenario 1: Generate a working program in a Python integrated development environment (IDE) such as Anaconda. The following example uses the Jupyter notebook program as part of the Anaconda Install.

Scenario 2: Expand code requiring 2 audio requests but deliver a single audio outcome file

Hint: The trick of this scenario is to create 2 separate myWords variables.

- In Python variables are either implicitly or explicitly declared.
- Code line 7 “my Words” is an implicit declaration as its type is not declared, such a character (char) or number
- Add a “_1” to the variable and then duplicate code lines 5-8 with a second variable myWords_2
- Finally, combine the myWords_1 with myWords_2 into myWords to deliver the audio output

```
""" Part 1: Set Computer File Directory os=operating system"""
import os
os.chdir('C:\\Users\\17574\\Desktop')

""" Part 2: Set Google Speech Recognition and Microphone Library Functions
import speech_recognition as sr
import pyaudio

""" Part 3: Ask user to same something use Google speech to parse words"""
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))

>>> Ready? Say something quick
>>> You Said...: Nacho

"""Part 4: Encode words into audio file audio data is binary so add 'wb'
                                for 'write binary data (1 or 0)"""
with open("myAudio.wav", "wb") as file_:
    file_.write(myWords.get_wav_data())

"""Part 5: Import a generic microphone module """
from playsound import playsound
playsound('myAudio.wav')

import os
os.chdir('C:\\Users\\17574\\Desktop')
import speech_recognition as sr
import pyaudio
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords_1 = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords_2 = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
myWords = myWords_1 + myWords_2
with open("myAudio.wav", "wb") as file_:
    file_.write(myWords.get_wav_data())
from playsound import playsound
playsound('myAudio.wav')

>>> Ready? Say something quick
>>> You Said...: Nacho
>>> Ready? Say something quick
>>> You Said...: Nacho

""" Run like a Pro """
import os
os.chdir('C:\\Users\\17574\\Desktop')
import speech_recognition as sr
import pyaudio
with sr.Microphone() as source:
    print("Ready? Say something quick")
    myWords = sr.Recognizer().listen(source)
    print("You Said...: "+ sr.Recognizer().recognize_google(myWords))
with open("myAudio.wav", "wb") as file_:
    file_.write(myWords.get_wav_data())
from playsound import playsound
playsound('myAudio.wav')

>>> Ready? Say something quick
>>> You Said...: I like cake
```