
CMSC 201 Spring 2019

Project 1 – Hawaiian Pronunciation

[Read this entire document before you begin!](#)

Assignment: Project 1 – Hawaiian Pronunciation

Due Date:

Design Document: Friday, March 29th, 2019 by 11:59:59 PM

Project: Friday, April 5th, 2019 by 11:59:59 PM

Value: 80 points

Collaboration: For Project 1, **collaboration is not allowed** – you must work individually. You may still come to office hours for help, but you may not work with any other CMSC 201 students.

Make sure that you have a complete file header comment at the top of each file, and that all of the information is correctly filled out.

```
# File:      FILENAME.py
# Author:    YOUR NAME
# Date:      THE DATE
# Section:   YOUR DISCUSSION SECTION NUMBER
# E-mail:    YOUR_EMAIL@umbc.edu
# Description:
#           DESCRIPTION OF WHAT THE PROGRAM DOES
```

This assignment is based on one created by Kendall Bingham for the University of Missouri - Kansas City, and presented at the SIGCSE 2019 Nifty Assignments session.

Instructions

For this project, you will be creating a single program, but one that is bigger in size and complexity than any individual homework problem. This assignment will focus on using functions to break a large task down into smaller parts. You will be required to turn in a design document before you turn in your actual code.

This is the first assignment where you've had to turn in a "design document" in addition to the actual code. The design document is intended to help you practice deliberately constructing your program and how it will work, rather than coding as you go along, or starting without a plan.

A list of functions required for this project are found later in this document. For Project 1, you are **required to follow the provided design exactly!** You may add additional functions, but you must implement all of the specified functions as described in the design overview.

**At the end, your Project 1 file must run without any errors.
It must also be called proj1.py (case sensitive).**

Additional Instructions – Creating the proj1 Directory

During the semester, you'll want to keep your different Python programs organized, organizing them in appropriately named folders (also known as directories).

You should create a directory in which to store your Project 1 files. We recommend calling it `proj1`, and creating it inside a newly-created directory called `Projects` inside the `201` directory.

If you need help on how to do this, refer back to the detailed instructions in Homework 1.

Objective

Project 1 is designed to give you lots (and lots) of practice with functions, as well as help you become more familiar and comfortable with “2D” lists (more specifically, lists of strings). You’ll need to use `while` loops, control statements like `if/else`, passing in parameters, returning from functions, shallow copies of lists, and algorithmic thinking.

Task

You’ll be implementing a program that provides a pronunciation for any valid Hawaiian phrase the user enters.

The program will need to check the validity of the phrases, and will need to follow all of the provided rules for how pronunciation works.

See the sample output at the end for detailed examples, and continue reading for more information.

Coding Standards

Prior to this assignment, you should re-read the Coding Standards, linked on the course website at the top of the “Assignments” page.

For now, you should pay special attention to the sections about:

- Comments
 - **Function header comments**
- Constants
 - You must use constants instead of magic numbers or strings!!!
- Make sure to read the last page of the Coding Standards document, which prohibits the use of certain tools and Python keywords.

Additional Specifications

For this assignment, **you must follow the design overview** in this document.

Project Guidelines

The project is worth a total of 80 points. Of those points 10 will be based on your design document, 10 will be based on following the coding standards, and the other 60 will be based on the functionality and completeness of your project.

Design Document

The design document will make sure that you begin thinking about your project in a serious way early on in the process. This will not only give you important experience doing design work, but will help you gauge the number of hours you'll need to set aside to be able to complete the project. **Your design document must be called design1.txt.**

For Project 1, the design overview is included in this document, and you are **required to follow it**. For future projects, you will be creating the design entirely on your own, and may choose to design it however you like.

Your design document must have four separate parts:

1. A file header, similar to those for your assignments
2. Constants
 - a. A list of all the constants your program will need, including a short comment describing what each “group” of constants is for
3. Function headers
 - a. A complete function header comment for each function
 - b. You do not need to include the function’s code in your design
4. Pseudocode for `main()`
 - a. A brief but descriptive breakdown of the steps your `main()` function will take to completely solve the problem; note function calls under the relevant comment (if applicable)

Although you will be presented with a design overview, you must still create the function headers and the pseudocode for `main()` on your own.

A start for your design is provided on Blackboard under “Assignments”. Follow the layout and format of that document. You can also copy it using:
`cp /afs/umbc.edu/users/k/k/k38/pub/cs201/design1.txt .`

NOTE: The sample design provided is not complete, and is missing many constants and function header comments. You must add them!

Your `design1.txt` file will be compared to the `proj1.py` file that you submit. Minor changes to the design are acceptable. A minor change might be the addition of another function, or a small change to `main()`.

Major changes between the design and your project will lose you points. This would indicate that you didn't give sufficient thought to your design.

(If your submitted design doesn't work, it is generally better to lose the points on the design, and to have a functional program, rather than turning in a broken program that follows the design. The decision is ultimately up to you.)

To submit your design document, use

```
linux1[4]% submit cs201 PROJ1_DESIGN design1.txt
Submitting design1.txt...OK
linux1[5]% █
```

Sample Output

The sample output is available at the end of this project description.

Other Hints

- Writing up and testing each of your functions individually will make your life much, much easier.
- Commenting your code as you write it or even before you write it (like writing pseudocode) will make the process much simpler, and will also allow the TAs to help you more effectively during the process.
- This project lends itself very well to a “top down” implementation, where functions are first made as dummy functions, and you ensure that everything works together. (Dr. Gibson’s first draft of the program just said “nice word choice, beep boop” or “how to say WORD” and similar things.)

Project 1: Hawaiian Pronunciation Rules

The Hawaiian language only contains 13 characters: eight consonants and five vowels. The eight consonants are h, k, l, m, n, p, w, and ' (a single apostrophe, called an *okina*). All Hawaiian words end in a vowel; the five vowels are a, e, i, o, and u.

When getting a phrase from the user, only these 13 characters (and the space character) are valid, and each word must end in a vowel.

The pronunciation of consonants is pretty simple: h, k, l, m, n, and p are pronounced just like they are in English. The w is a bit trickier – if it's at the start of a word, or follows an o or u, it's pronounced as a "w". But if it follows an a, e, or i, it's pronounced as a "v" instead. (We'll cover the okina in a bit.)

The pronunciation of vowels seems pretty simple at first, with each vowel having a specific pronunciation:

- | | | | | |
|-----|-------------|----|--------------|---------|
| • a | sounds like | ah | e.g., Austin | ah-stin |
| • e | sounds like | eh | e.g., egg | eh-gg |
| • i | sounds like | ee | e.g., bee | |
| • o | sounds like | oh | e.g., obey | oh-bay |
| • u | sounds like | oo | e.g., moo | |

These rules alone let us pronounce a number of simple Hawaiian words:

- aloha sounds like ah-loh-hah
- ohana sounds like oh-hah-nah
- wahine sounds like wah-hee-neh

If this was it for the pronunciation, we could simply check how to pronounce the "w"s, and then replace all the "a"s with "ah"s and all the "u"s with "oo"s and so on, and we would be done! But that can't be all, since that would mean that "Hawaii" was pronounced "hah-vah-ee-ee", that the island of "Maui" was pronounced "mah-oo-ee", and that "lei" was pronounced "leh-ee", none of which is right.

Some pairs of vowels are pronounced differently from their two components:

- ai sounds like eye
- ae sounds like eye *same as ai*
- ao sounds like ow *e.g., how*
- au sounds like ow *same as ao*
- ei sounds like ay *e.g., hay*
- eu sounds like eh-oo
- iu sounds like ew *e.g., dew*
- oi sounds like oy *e.g., boy*
- ou sounds like ow *same as ao*
- ui sounds like ooey *e.g., gooey*

Following these rules, we can “fix” the incorrect pronunciations from earlier:

- Hawaii *actually* sounds like hah-veye-ee
- Maui *actually* sounds like mow-ee
- lei *actually* sounds like lay

You’ll notice that *not* every vowel combination is represented: for example, there is no “oa” combined vowel sound. This means that the island “Oahu” is pronounced “oh-ah-hoo” with the “o” and “a” being pronounced separately.

There are also words with more than 2 vowels in a row, and these need to be handled from left to right. For example, the word *kaiāola* has four vowels in a row; since “ai” and “ao” are both combined vowel sounds, it’s pronounced “keye-ow-lah”. The word “kaiōke” has three vowels in a row, and is pronounced “keye-oh-keh”.

However, if there are more than 2 vowels in a row, and the first two vowels do not have a combined vowel sound, you pronounce the first vowel normally, and then check to see if the next two vowels have a combined sound. So the word “Huaai” would be pronounced “hoo-ah-eye” with the “u” and the first “a” being pronounced separately, since there is no “ua” or “aa” sounds (but there is an “ai” sound, hence the “eye” at the end).

Finally, [the okina](#), which indicates a “glottal stop” in a word. The okina doesn’t have a pronunciation – it stays as it is (a single apostrophe), but it can be used to separate vowels that would otherwise be pronounced together. For example, “Waha'ula” has an okina, separating the “au” into an “a'u” instead.

With the okina, it’s pronounced “wah-hah'oo-lah”.

Without the okina, it’s pronounced “wah-how-lah”.

Additional Details

- You must have the hyphens in the pronunciation guide, as is shown in the sample output and throughout the examples.
- Spaces must be kept intact when pronouncing a phrase.
- You are **NOT ALLOWED** to use the `.replace()` method or any other methods we have not covered. Use of any of these methods will lose you at least 10% off the final project grade.

Tips and Tricks

- Don’t worry about the exact placement of hyphens until everything else is working. (It’s generally easier to have more hyphens than needed, and remove the ones you don’t need.) The words are still readable if they have extra hyphens in them.
- The methods `.upper()` and `.lower()` are very helpful for simplifying dealing with capitalization.
- String slicing can also be a very helpful tool in this project.
- Constants that are lists can be easily used to hold information about pronunciation.

Design Information

You are required to implement and use at least the following seven functions for Project 1, in addition to `main()`. You may implement more functions if you think they are necessary, but the seven below must be implemented and used, or you will lose significant points on your project grade. The information for each function is below.

Printing Functions

As the name suggests, these functions print information to the user. **None** of them return values – they only print information, which does NOT count as output. Some of them take in parameters, while others do not.

Printing functions are often an important part of a complex program, but Project 1 actually does not contain any! You'll normally encounter them when it's necessary to display menus for the user, or to print out things like game boards or the results of a search.

Helper Functions

As the name suggests, these functions “help” other functions, by performing small tasks that will be needed by many pieces of the program. They are often called from functions other than `main()`.

- **def getChoice(question, options)**
 - Gets a valid choice from the user, only allowing them to select one of the choices contained in `options`, using the provided string of `question` when asking the user for input
 - The choices should be accepted case **insensitive!** (So “YES” is fine, as is “Yes” and “yeS” and “yes”)
 - Takes in a string and a list of strings
 - The question is used by the function when asking for input
 - The list of strings contains all of the valid choices available
 - Returns a string

- **def getHawaiianPhrase ()**
 - Asks the user to input a Hawaiian phrase, re-prompting them if the phrase is not valid (and explaining why that is the case)
 - Function takes in no input
 - Returns a string of the user's chosen Hawaiian phrase

- **def pronounceW(word, index)**
 - Determines how a "w" should be pronounced
 - Takes in a string and an integer
 - The string is the word the "w" exists in
 - The integer is the index where the "w" resides
 - Returns a single character string
 - Either "v" or "w" depending on the pronunciation

- **def simpleVowel(letter)**
 - Determines how a vowel is pronounced
 - Takes in a single-character string
 - Returns a string, how the vowel is pronounced

- **def complexVowel(vowels)**
 - Determines how a complex, two-letter vowel is pronounced
 - Takes in a two-character string
 - Returns a string, how the complex vowel is pronounced

General Functions

These are the “heavy lifters” of the program, and do the majority of the work, and are generally called from `main()`. They often call other functions, often more than one.

- **def pronounce (phrase)**
 - Determines how to pronounce an entire phrase
 - Takes a string, possibly with multiple words
 - Returns a string containing the complete phrase’s pronunciation

- **def pronounceWord (word)**
 - Determines how to pronounce a single word
 - Takes in a string (with no spaces)
 - Returns a string containing the word’s pronunciation
 - **WARNING:** This function is probably the most complex one in the whole project, so take some time to think carefully about what it needs to do and plan how it will accomplish that before you start writing any code for it.

Sample Output: Input Validation

(Yours does not have to match this word for word, but it should be similar.)

```

linux1[10]% python3 proj1.py
Enter a Hawaiian phrase to pronounce: invalid
The word invalid does not end in a vowel
The letter v is not valid
The letter d is not valid
Enter a Hawaiian phrase to pronounce: ABCDEFG hijklmn !op
The word ABCDEFG does not end in a vowel
The word hijklmn does not end in a vowel
The word !op does not end in a vowel
The letter B is not valid
The letter C is not valid
The letter D is not valid
The letter F is not valid
The letter G is not valid
The letter j is not valid
The letter ! is not valid
Enter a Hawaiian phrase to pronounce: Moana of Matanui
The word of does not end in a vowel
The letter f is not valid
The letter t is not valid
Enter a Hawaiian phrase to pronounce: Moana
The phrase MOANA is pronounced:
    moh-ah-nah
Do you want to enter another word? (y/yes/n/no): moana
Enter y/yes/n/no
Do you want to enter another word? (y/yes/n/no): YES
Enter a Hawaiian phrase to pronounce: aloha
The phrase ALOHA is pronounced:
    ah-loh-hah
Do you want to enter another word? (y/yes/n/no): y
Enter a Hawaiian phrase to pronounce: waikiki
The phrase WAIKIKI is pronounced:
    weye-kee-kee
Do you want to enter another word? (y/yes/n/no): NO

```

Sample Output: Simple Words and Phrases

(Yours does not have to match this word for word, but it should be similar.)

```
linux1[11]% python3 proj1.py
Enter a Hawaiian phrase to pronounce: poke
The phrase POKE is pronounced:
    poh-keh
Do you want to enter another word? (y/yes/n/no): yes
Enter a Hawaiian phrase to pronounce: Mahalo
The phrase MAHALO is pronounced:
    mah-hah-loh
Do you want to enter another word? (y/yes/n/no): y
Enter a Hawaiian phrase to pronounce: aloha oe
The phrase ALOHA OE is pronounced:
    ah-loh-hah oh-eh
Do you want to enter another word? (y/yes/n/no): y
Enter a Hawaiian phrase to pronounce: humuhumunukunukuapuaa
The phrase HUMUHUMUNUKUNUKUAPUAA is pronounced:
    hoo-moo-hoo-moo-noo-koo-noo-koo-ah-poo-ah-ah
Do you want to enter another word? (y/yes/n/no): YES
Enter a Hawaiian phrase to pronounce: Papalealii
The phrase PAPALEARLII is pronounced:
    pah-pah-leh-ah-lee-ee
Do you want to enter another word? (y/yes/n/no): Yes
Enter a Hawaiian phrase to pronounce: momoa
The phrase MOMOA is pronounced:
    moh-moh-ah
Do you want to enter another word? (y/yes/n/no): nope
Enter y/yes/n/no
Do you want to enter another word? (y/yes/n/no): no
```

Sample Output: Complex Words and Phrases

(Yours does not have to match this word for word, but it should be similar.)

```
linux1[12]% python3 proj1.py
Enter a Hawaiian phrase to pronounce: lanai
The phrase LANAI is pronounced:
    lah-neye
Do you want to enter another word? (y/yes/n/no): y
Enter a Hawaiian phrase to pronounce: keiki
The phrase KEIKI is pronounced:
    kay-kee
Do you want to enter another word? (y/yes/n/no): yes
Enter a Hawaiian phrase to pronounce: Kahiaka
The phrase KAHIAKA is pronounced:
    kah-hee-ah-kah
Do you want to enter another word? (y/yes/n/no): yes
Enter a Hawaiian phrase to pronounce: Alaolii
The phrase ALAOLII is pronounced:
    ah-low-lee-ee
Do you want to enter another word? (y/yes/n/no): Y
Enter a Hawaiian phrase to pronounce: Kaahele
The phrase KAAHELE is pronounced:
    kah-ah-heh-leh
Do you want to enter another word? (y/yes/n/no): y
Enter a Hawaiian phrase to pronounce: Ho'olaule'a
The phrase HO'OLAULE'A is pronounced:
    hoh'oh-low-leh'ah
Do you want to enter another word? (y/yes/n/no): nO
```

Sample Output: Interesting Words

(Yours does not have to match this word for word, but it should be similar.)
(If you want to learn something, try looking up some of these words!)

```
linux1[13]% python3 proj1.py
Enter a Hawaiian phrase to pronounce: Kamakawiwo'ole
The phrase KAMAKAWIWO'OLE is pronounced:
    kah-mah-kah-vee-voh'oh-leh
Do you want to enter another word? (y/yes/n/no): YES
Enter a Hawaiian phrase to pronounce: Aiea
The phrase AIEA is pronounced:
    eye-eh-ah
Do you want to enter another word? (y/yes/n/no): yes
Enter a Hawaiian phrase to pronounce: Lili'uokalani
The phrase LILI'UOKALANI is pronounced:
    lee-lee'oo-oh-kah-lah-nee
Do you want to enter another word? (y/yes/n/no): y
Enter a Hawaiian phrase to pronounce: Kahanamoku
The phrase KAHANAMOKU is pronounced:
    kah-hah-nah-moh-koo
Do you want to enter another word? (y/yes/n/no): yes
Enter a Hawaiian phrase to pronounce: Kamehameha
The phrase KAMEHAMEHA is pronounced:
    kah-meh-hah-meh-hah
Enter a Hawaiian phrase to pronounce: E ke onaona noho i ka
lipo
The phrase E KE ONAONA NOHO I KA LIPO is pronounced:
    eh keh oh-now-nah noh-hoh ee kah lee-poh
Do you want to enter another word? (y/yes/n/no): NO
```

Helpful Checklist

We understand that the first big project can be overwhelming, which is why we've provided you with a detailed description of all the functions, along with sample output, a starter design, and more. Below, you'll find a summary of the key actions and where important documents can be found.

Design:

- Download the starter design from Dr. Gibson's public directory:
 - `cp /afs/umbc.edu/users/k/k/k38/pub/cs201/design1.txt .`
- Read through the function descriptions provided in this document, and create function header comments for each function in your design
- Look at the sample output, and begin sketching out an idea in pseudocode for how your `main()` will work
 - Look at the function headers you wrote to gain more insight

Coding:

- [Optional step] Make a copy of your design (or your design so far), as a framework for starting the project
- Going off of your design, start coding up your project piece by piece
 - Use top-down or bottom-up implementation – your choice!
 - Don't code up everything at once, and test as you go
- If you make changes to your design while coding, and it's before the design due date, update your design and resubmit it!
- Come to office hours when you need help or get stuck

General:

- Submit the design to PROJ1_DESIGN, **not** PROJ1
- Start working on the project itself before March 29th if possible, since actually coding it up will improve your design as you go
- Test your code with the user choices made in the sample output

Submitting

Once your `proj1.py` or `design1.txt` file is complete, it is time to turn it in with the `submit` command. (You may also turn the project in multiple times, as you reach new milestones. To do so, run `submit` as normal.)

To submit your design file (which is due Friday, March 29th, 2019 by 11:59:59 PM), use the command:

```
linux1[4]% submit cs201 PROJ1_DESIGN design1.txt
Submitting design1.txt...OK
linux1[5]% █
```

To submit your project file (which is due Friday, April 5th, 2019 by 11:59:59 PM), use the command:

```
linux1[4]% submit cs201 PROJ1 proj1.py
Submitting proj1.py...OK
linux1[5]% █
```

If you don't get a confirmation like the one above, check that you have not made any typos or errors in the command.

You can check that your assignment was submitted by following the directions in Homework 0. Double-check that you submitted your homework correctly, since **an empty file will result in a grade of zero for this assignment.**