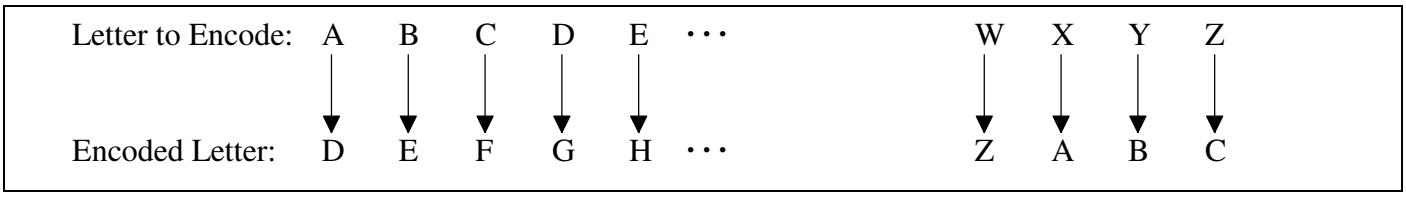
Homework #1 Data Structures — Due: Sept. 6 (Saturday at 11:59 PM)

File Encryption/Decryption Program: You are to **design** and write a menu-driven program that allows users to interactively select between:

- encrypting a user-specified text-file (`.txt` extension) using a simple Caesar-cipher encryption scheme to generate a new encoded file with the same name, except with a `.zzz` extension. The Caesar cipher encryption scheme substitutes each letter in a message by another letter some fixed number of positions down (with wrap around) the alphabet. For example, a shift by 3 gives the following substitutions:



  Thus, a line of text "Meet by Union!" with a shift of 3 would be encoded as "Phhw eb Xqlrq!". Non-letter characters (white-space, punctuation, etc.) should just be copied directly to the encoded file.
- decrypting a user-specified Caesar-cipher encrypted file (`.zzz` extension) to generate a new text-file with the same name, except with a `.txt` extension.

Your program also needs to: (Download the `hw1.zip` sample programs)

- validate a correct menu selection: encrypt, decrypt, or exit.
- validate that the user specified file to encrypt/decrypt exists and force the user to reenter until they specify an existing file. Feel free to make the program more user-friendly by listing all files of the appropriate type `.txt` on encryption or `.zzz` on decryption.
- validate the user specified shift amount is a positive integer (force the user to reenter until a positive integer is entered)
- check if the user specified file name for the encoded file exists before opening it for write. If it already exists, ask the user if they are okay with it being wiped out. If they are not, ask them to pick a different file name to receive the encoded text.

Your program's interaction should looks something like: (Student input shown in **bold**.)

```
Welcome to the Caesar Cipher Encryption/Decryption Program

Would you like (e)ncrypt a file, (d)ecrypt a file, or e(x)it (enter e, d, or x)? e

Enter the text-file name to encrypt: mesage.txt
Sorry the file 'mesage.txt' does NOT exist -- please try again!
Enter the text-file name to encrypt: messaeg.txt
Sorry the file 'messaeg.txt' does NOT exist--please try again!
Enter the text-file name to encrypt: message.txt

Enter the shift amount for encoding:  three
The shift amount must be a positive integer (e.g., 5) -- please try again!
Enter the shift amount for encoding:  3

WARNING:  The file 'message.zzz' already exists!
Is it okay to wipe it out (y/n)?  y

The file 'message.txt' was successfully encrypted with a shift amount of 3 to file 'message.zzz'
```

Save your program in a file called caesarCipher.py

When you write your program, be sure you:

- think about the functional-decomposition (top-down) design **before you start to write code**! You will need to turn in a design document (see below for details).
- use meaningful variable names with good style (i.e., useCamelCase)
- use comments (""" Multi-line Comment """) at the start of the program **and** immediately after each function `def`inition describing what they do (see lab1 `diceOutcomes.py` program for an example)

- use a main function (see lab1 `diceOutcomes.py` program) located at the top of program with a call to `main()` at the bottom of the file to start execution
- use global constants where appropriate with good style (ALL_CAPS_AND_UNDERSCORES). (Put your global constants after your initial comments describing the program and before your main function definition so they can be found and changed easily in future versions of your program.)
- allow the user to enter the file name to be analyzed and verify that the file name exsists (`import os.path`)

Submit the single file, hw1.zip containing the following:

- **caesarCipher.py** (your Python program)
- **design.doc** (or design.pdf, or design.txt, or design.rtf) a document describing the design of your program including a functional-decomposition diagram with text describing each function (see lab1 description)

The steps for the homework submission system are:

1. Design, write, debug, and test your program in the hw1 folder. When you are ready to submit your homework, zip the whole folder by right-clicking on it and selecting `Send to | Compressed (zipped) folder`. This will create a new file called `hw1.zip` which you will submit electronically. (This assumes Windows OS....)
2. Log on to the submission system at: **https://www.cs.uni.edu/~schafer/submit/which_course.cgi**

   (It is very likely that you will get some security certificate warnings when trying to use this. You may add an exception and accept the existing security certificate.) Use the same CatID user-name and password you use to log on the lab computers.
3. Select the course and section number of "CS 1520, Data Structures, Fienup". Click the "Continue".
4. Select the homework that you wish to submit: "HW 1: Caesar Cipher". Click the "Continue" button.
5. Specify how many extra files you want to submit. Just leave it at 0. Click the "Continue" button.
6. Upload your program by Browsing and selecting your `hw1.zip` file. Click the "Continue" button.
7. The next page reports on the status of the upload(s). You can always continue to upload a better version of the program until the deadline. The newer file will replace an older file of the same name.

(If you miss the deadline, you'll need to submit it as above, but select "Late Homeworks" in step 4 above.)