

# Introduction to Programming in Python

## Final Project

**Due: November 15, 2020 by 11:59pm**

Some of you might be familiar with the movie “Imitation Game.” The movie focused on the life of Alan Turing. He is one of the most important people in the history of computer science. He interpreted encrypted messages during the Second World War. Some would argue that his interpretation of encrypted messages helped end the war early.

For this project, you will do something similar to what Alan Turing did in the 1940s. Since you are just learning Python, you are going to implement very simple encryption and decryption techniques.

### What to do?

- You will write two classes.
  - One class should be used to read from files and write to files. This class should contain two or more methods.
    - \* At least one method should receive *input\_filename* as a parameter, read from a text file, and return a string.
    - \* At least one method should receive a *string* and *output\_filename* as parameters, and print the string to the output file.
  - Another class that does encryption and decryption of text. This class should contain two or more methods. One method to encrypt text, another to decrypt text, and more if needed.
- There are no limitations on the number of classes and methods. You are free to create more classes and methods if you need. However, put each class in a separate file to use as a module.
- Once the modules are created, create the *main.py* file. This file should do the following -
  - Give the user the following choices -
    1. Type 1 to encrypt
    2. Type 2 to decrypt
    3. Type anything else to exit
  - Once the user enters a choice, if the choice is **1**, your program should give a prompt to the user to enter the name of a text file to encrypt. Once done, it should add the extension *\_\_encrypted* to the file name.

- \* For example, if the filename entered is *engima.txt*, the output filename should be *engima\_\_encrypted.txt*.
- If the choice entered is 2, then your program should give prompt to enter the filename to decrypt. Your program should first decrypt the text. Then it should check if the filename contains the word *encrypted* in it. If it does, it should be changed to *decrypted*. If it doesn't, then the extension *\_\_decrypted* should be added to it. This modified filename becomes the output filename.
- \* For example, if the user enters *engima\_\_encrypted.txt*, then output filename becomes *engima\_\_decrypted.txt*. But if the user simply enters *engima.txt*, then it should be changed to *engima\_\_decrypted.txt*.

### What to know?

- Encryption
  - One of the easiest ways to encrypt text is to take the ASCII equivalent value of a character and add 6 to it. This will change the representations of **a** to **g**, **b** to **h** and so on.
  - Take a look at ASCII chart clicking [here](#).
  - If you encounter symbols and numbers, e.g. \$, @, #, 1, 23 etc. , you may leave them as they are.
  - However, if you look at the ASCII chart, and complete the above three steps, you will find that adding 6 to {U/u, V/v, W/w, Y/y, Z/z} may make the encrypted data undecipherable. For these special cases, that is for {U-Z} and {u-z}, subtract 20 instead.
- Decryption
  - Do the opposite of encryption.

### Examples

- successful encryption -
  - original text
 

The lazy brown dog!!!
  - encrypted text -
 

Znk rgfe hxuct jum!!!
- successful decryption -
  - encrypted text
 

Znk rgfe hxuct jum!!!

– decrypted text -

The lazy brown dog!!!

### **Submission Notes**

- You may work alone or in a group (maximum two people). If you prefer to work in a group, please notify me by 11:59pm on November 5. If you are working in a group, send me one email informing me who your partner is (one email for one group and CC your partner). If you prefer working alone, please send me an email informing me that.
- Put all your files in a folder and zip the folder. Use your first-Name\_lastName as the name of the zipped folder.
- No late submissions will be accepted.

### **Useful functions**

- `ord()`
- `chr()`