Programming Exercise: Object Oriented Caesar Cipher

Assignment 1: One Key

In this assignment, you will put together the **CaesarCipher** class from the lesson and add a **decrypt** method to decrypt with the same key. In addition you will create a second class, **TestCaesarCipher** to test examples that use the **CaesarCipher** class, including writing a method that will automatically decrypt an encrypted file by determining the key and then decrypting with that key.

Specifically, you should do the following.

- Create the **CaesarCipher** class with the following parts:
    - Private fields for the **alphabet** and **shiftedAlphabet**
    - Write a constructor **CaesarCipher** that has one int parameter **key**. This method should initialize all the private fields of the class.
    - Write an **encrypt** method that has one String parameter named **input**. This method returns a String that is the input encrypted using **shiftedAlphabet**.
    - Write a **decrypt** method that has one String parameter named **input**. This method returns a String that is the encrypted String decrypted using the key associated with this **CaesarCipher** object. One way to do this is to create another private field **mainKey**, which is initialized to be the value of **key**. Then you can create a **CaesarCipher** object within **decrypt**:

      ```
      CaesarCipher cc = new CaesarCipher(26 - mainKey);
      ```

      and call `cc.encrypt(input)`.
- Create the **TestCaesarCipher** class with the following parts:
    - Include the methods **countLetters** and **maxIndex** that you wrote in the previous lesson.
    - Write the void method **simpleTests** that has no parameters. This method should
        - read in a file as a String
        - create a **CaesarCipher** object with key 18

        - encrypt the String read in using the **CaesarCipher** object
        - print the encrypted String
        - decrypt the encrypted String using the **decrypt** method.
    - Write the method **breakCaesarCipher** that has one String parameter named **input**. This method should figure out which key was used to encrypt this message (in a similar manner as the previous lesson), then create a **CaesarCipher** object with that key and decrypt the message.
    - In the **simpleTests** method, add a call to **breakCaesarCipher** on the encrypted String to decrypt it automatically by determining the key, and print the decrypted String.

Assignment 2: Two Keys

In this assignment, you will put together the **CaesarCipherTwo** class that encrypts a message with two keys (the same way as the previous lesson: **key1** is used to encrypt every other letter, starting with the first, and **key2** is used to encrypt every other letter, starting with the second), and also decrypts the same message. In addition you will create a second class, **TestCaesarCipherTwo** to test examples that use the **CaesarCipherTwo** class, including writing a method that will automatically decrypt an encrypted file by determining the two keys that were used to encrypt it.

Specifically, you should do the following.

- Create the **CaesarCipherTwo** class with the following parts:
    - Include private fields for the **alphabet**, **shiftedAlphabet1**, and **shiftedAlphabet2**.
    - Write a constructor **CaesarCipherTwo** that has two int parameters **key1** and **key2**. This method should initialize all the private fields.
    - Write an **encrypt** method that has one String parameter named **input**. This method returns a String that is the input encrypted using the two shifted alphabets.
    - Write a **decrypt** method that has one String parameter named **input**. This method returns a String that is the encrypted String decrypted using the **key1** and **key2** associated with this **CaesarCipherTwo** object. You might want to add more private fields to the class.
- Create the **TestCaesarCipherTwo** class with the following parts:
    - Include the methods **halfOfString**, **countLetters**, and **maxIndex** that you wrote in the previous lesson.
    - Write the void method **simpleTests** that has no parameters. This method should
        - read in a file as a String
        - create a **CaesarCipherTwo** object with keys 17 and 3
        - encrypt the String using the **CaesarCipherTwo** object

        - print the encrypted String
        - decrypt the encrypted String using the **decrypt** method.
    - Write the method **breakCaesarCipher** that has one String parameter named **input**. This method should figure out which keys were used to encrypt this message (in a similar manner as before), then create a **CaesarCipherTwo** object with that key and decrypt the message.
    - In the **simpleTests** method, add a call to **breakCaesarCipher** on the encrypted String to decrypt it automatically by determining the keys and then print the decrypted String.