Lab #5: Messaging System

*Carson Kramer*

*March 28th, 2023*

ALGORITHM

1. Main menu- prompts for encrypt, decrypt, or quit

2. Prompt the user for the password

3. Prompt the user for the name of the file

4. Print out the text from the file (either plaintext or the encrypted message)

5. Prompts for a name of the new file to be created

6. Prints out the final message (both to the console and to the new file)

Tokenization

* declare a character array that will hold 1000 characters in main
* getline command here to read the contents of the file into this array
* Read the contents of the file into a character array
* Add a null terminator at the end of the string to stop from printing characters beyond what was in the file
* Use encrypt / decrypt functions to add individual words to tokenization vector in message instance

Vigenère Class

* Public:
  + One-argument constructor that receives a string for the encryption key.
    - Call capitalization function in constructor
  + Get and set functions
  + encrypt and decrypt a single word functions (string)
* Private:
  + String encryption key - called “key”
  + function to convert a string entirely to uppercase
    - Loop with toupper function for each character

Message Class

* Middle man between client and Vigenère Class
* No math actually completed in Message Class
* Public:
  + One-argument constructor that brings in the string key
  + void showWords()
    - Print vector of encrypted or decrypted words
  + makeFile(string fileName)
  + encryptWord()
    - function should receive the token as a char\* data type and cast it into a string variable
    - push that newly encrypted word onto the vector of words.
  + encrypt()
  + decryptWord()
  + decrypt()
* Private:
  + vector of all of your tokenized words
  + private data member of the Vigenère class (demonstrating composition)

SCREEN-SHOTS OF RUNNING PROGRAM

Text

Description automatically generated

INTEGRITY STATEMENTS

* I have not shared the source code in my program with anyone other than the pre-approved human sources.
  + *Please include a note here if you have used one or more of the pre-approved human sources or received special permission from me.*
* I have not used source code obtained from another student, or any other unauthorized source, either modified or unmodified.
* If any source code or documentation used in my program was obtained from another source, such as the course textbook or course notes, that has been clearly noted with a proper citation in the comments of my program.
  + It would also be helpful to include a note here of which sources you used
* I have not knowingly designed this program in such a way as to defeat or interfere with the normal operation of any machine it is graded on or to produce apparently correct results when in fact it does not.

Note: These statements serve as your personal promise that the above is true. If I find that you have not been true to ALL of the four statements above, you will get a zero for the assignment and receive an academic violation report (which goes on your academic record). Both are minor compared to the loss of your integrity.