COM 252 OOP
Final Exam
Due: Official Final Exam Date (UBIS System)

You will design the following application for your final exam:

1) Design a **Encrypter class** with the following specifications:

a. Has a list of references to encryption algorithms.

b. Has a list of references to its listeners (those listening for encrypted messages from the Encrypter class), there are two listener classes for this project (General and Spy described below).

c. Has a method called register that takes three arguments: a reference to an encryption algorithm, the key that should be used with the encryption algorithm and a reference to a listener. Listener is supposed to call this method passing the encryption algorithm reference, the key and reference to itself (the listener).

When the Encrypter object wants to send a message (**to all its listeners**), it first obtains the plain text from a file given to it and encrypts the message in it using an encryption algorithm. The same plain text message is encrypted using different encryption algorithms (for this project you only have to implement two). Each encrypted message is sent to "its" listener using "its" listener's encryption algorithm (passed initially by its listeners via the register method).

2) Design the following two classes that listen for the encrypted messages from an Encrypter object. **Note that you should be able to add more listeners (of different types) without having to change your codebase.**

a. **General class**. General class as stated is a listener for the Encrypter class. Assume that an object of type General passes a reference to the Shift Encryption algorithm (described below) when it registers itself with the Encrypter object.

b. **Spy class**. Spy class as stated is a listener for the Encrypter class. Assume that an object of type Spy passes a reference to the BinaryEncryption algorithm (described below) when it registers itself with the Encrypter object.

Both Spy and General classes have a method called update(String s) that takes an encrypted message and can decrypt the encrypted message sent by the Encrypter object via the update method (called on listeners).

3) Have an **interface** called **Encyrption** with one method called **String encrypt(String plainText, byte key)** that is implemented by two concrete encryption algorithms described below. Encrypter class has a list of encryption algorithms of type Encryption. Every time a listener (General or Spy) registers with Encrypter object, the reference to the encryption algorithm is passed by the listener as an argument to the register method of Encrypt object and added to the list of type Encryption in the Encrypter object.

4) Have an **interface** called **Decryption** with one method called **String decrypt(String encryptedText, byte key)** that is implemented by two concrete decryption algorithms described below. General and Spy (listeners for the Encrypt) classes have a reference to a decryption algorithm of type Decryption. Every time a listener (General or Spy) needs to decrypt a message, they will call their decrypt method that returns the plain text.

The following are the two encryption algorithm that you would implement:

Shift Encryption: Given a text (of English with nothing else other than text, that is, text consisting of only alphabetic characters), your algorithm should shift every letter “right” in the plain text by a randomly chosen number between 1 or 25 places (the key). For example if the plain text is “the attack will start at five am” and the randomly chosen key is 1 then the encrypted message would be “uif buubdl xjmm tubsu bu gjwf bn”. The same key is used by the decryption algorithm. Decryption should shift the encrypted message “left” “key many places”.

Binary Encryption: The message file is opened in binary mode. Each byte is encrypted using the key passed applying the bitwise XOR operation (between a byte of the plain text and the key). Remember that the key that should be used to encrypt the message is passed via the register method called on the Encrypter object.
Assume that the first line contains the first 4 bytes of the plain message (before encryption) in binary. The second line is the key (a byte long key which is repeated for every byte of the plain text). We XOR each byte of the plain text with the key so that we get the third line below (the first four bytes of the encrypted message).

01010111 01101001 01101011 01101001 plain text (first 4 bytes)
11110011 11110011 11110011 11110011 byte-long key
10100100 10011010 10011000 10011010 encrypted message

This all the encryption algorithm will do. On the decryption side we will apply the same XOR operation to every byte of the encrypted message together with the key to obtain the plain text message.

10100100 10011010 10011000 10011010 decrypted message
11110011 11110011 11110011 11110011 byte-long (the same) key
01010111 01101001 01101011 01101001 plain text

Note that the same key is used both by the encryption and decryption algorithms in both methods.

Deliverables:
1. Your project file with a readme.txt file. The readme.txt file should contain what you have accomplished and what you could not. Explain anything you think may help me understand your work.
2. Output of your application that gives me an idea of a run of your program.