CSE1322 Assignment 2

Background:

For this assignment, you will code a simple encryption application. The application encrypts words by mapping a list of the twenty-six alphabetic characters to a list of twenty-six corresponding symbols using the lists indices.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
!	@	#	$	^	&	*	(	)	_	-	+	=	?	,	{	}	[	]	/	|	;	:	.	<	>

symbols list

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

alphabets list

For example, the word Elvis, would encrypt as follows:

E -> ^
l -> +
v -> ;
i ->)
s ->]

So, Elvis would encrypt to: ^+;)]

Your task:

You will start with the following class definition:

C#	Java

using System;
using System.Collections.Generic;
 	
class Encryption
{
  List<char> symbols = new List<char>();
  List<char> alphabets = new List<char>();
 	
  public Encryption()
	import java.util.Scanner;
	import java.util.ArrayList;
	
	class Encryption
	{
	  ArrayList<Character> symbols = new ArrayList<Character>();
	  ArrayList<Character> alphabets = new ArrayList<Character>();
	
	  public Encryption()

```
 {
  symbols.Add('!');
  symbols.Add('@');
  symbols.Add('#');
  symbols.Add('$');
  symbols.Add('^');
  symbols.Add('&');
  symbols.Add('*');
  symbols.Add('(');
  symbols.Add(')');
  symbols.Add('_');
  symbols.Add('-');
  symbols.Add('+');
  symbols.Add('=');
  symbols.Add('?');
  symbols.Add(',');
  symbols.Add('{');
  symbols.Add('}');
  symbols.Add('[');
  symbols.Add(']');
  symbols.Add('/');
  symbols.Add('|');
  symbols.Add(';');
  symbols.Add(':');
  symbols.Add('.');
  symbols.Add('<');
  symbols.Add('>');

  for(char letter='a';letter<='z';letter++)
  {
   alphabets.Add(letter);
  }
 }
}
```

```
 {
  symbols.add('!');
  symbols.add('@');
  symbols.add('#');
  symbols.add('$');
  symbols.add('^');
  symbols.add('&');
  symbols.add('*');
  symbols.add('(');
  symbols.add(')');
  symbols.add('_');
  symbols.add('-');
  symbols.add('+');
  symbols.add('=');
  symbols.add('?');
  symbols.add(',');
  symbols.add('{');
  symbols.add('}');
  symbols.add('[');
  symbols.add(']');
  symbols.add('/');
  symbols.add('|');
  symbols.add(';');
  symbols.add(':');
  symbols.add('.');
  symbols.add('<');
  symbols.add('>');

  for(char letter='a';letter<='z';letter++)
  {
   alphabets.add(letter);
  }
 }
}
```

This creates two lists (2 ArrayLists in Java, and 2 Lists in C#). The first list contains twenty-six symbols (i.e. ~,@,#,$,%, etc.), one symbol in each cell. The second list contains the twenty-six lower case letters of the alphabet (a – z), one letter in each cell. From there you'll create the following six (6) methods:

1) Add a method return_alphabet, which takes in an int (integer) and returns the alphabet stored at that position, i.e. 5 would return f

2) Add a method return_alphabet_index, which takes in an alphabetic character (char) and returns the index (int) of the character in the alphabets list, i.e. a would return 0, b would return 1.
3) Add a method return_symbol, which takes in an int (integer) and returns the symbol stored at that position, i.e. 5 would return &
4) Add a method return_symbol_index, which takes in a symbol (char) and returns the index (int) of the symbol in the symbols list, i.e. ! would return 0, @ would return 1.
5) Add a method encrypt_message, which takes in a plain-text string and returns the encrypted version of that string, i.e. Dwags would return $:!*]
   - The method should convert the plain-text string to lowercase (hint: Java: .toLowerCase(), C#: ToLower())
   - The method should process each character in the plain-text string individually, encrypting each character and building a new string of encrypted characters (hint: Java: toCharArray(), C#: ToCharArray(), return_alphabet_index(), return_symbol())
   - If an invalid alphabet character is found, the following string should be returned: “Error: Invalid Character”
6) Add a method decrypt_message, which takes in an encrypted string and returns the decrypted version of that string, i.e. $:!*] would return dwags
   - The method should process each symbol in the encrypted string individually, decrypting each symbol and building a new string of decrypted characters (hint: Java: toCharArray(), C#: ToCharArray(), return_symbol_index(), return_alphabet())
   - If an invalid symbol is found, the following string should be returned: “Error: Invalid Symbol”

Driver Program:

Create an object of the class Encryption.

Prompt the user with the following menu:

1 Encrypt a message
2 Decrypt a message
3 Quit

Enter Choice

If the user enters 1, prompt the user to enter a message, pass the result to the encrypt_message method, and print the returned encrypted string.

If the user enters 2, prompt the user to enter an encrypted message, pass the encrypted message to the decrypt_message method, and print the returned decrypted string.

If the user enters 3, terminate the program.

If the user enters any character other than a 1, 2, or 3, the following error message should display: Error: Please enter valid input, and the user should be allowed to reenter a valid choice.

Sample Output:

```
1  Encrypt a message
2  Decrypt a message
3  Quit

Enter Choice:
1
Enter the plain text message:
MayTheForceBeWithYou

Encrypted Msg: =!</(^&,[#^@^:)/(<,|

1  Encrypt a message
2  Decrypt a message
3  Quit

Enter Choice:
2
Enter the encrypted message:
=!</(^&,[#^@^:)/(<,|

Decrypted Msg: maytheforcebewithyou

1  Encrypt a message
2  Decrypt a message
3  Quit
```

Enter Choice:
3

Submitting your answer:

Please follow the posted submission guidelines here:
https://ccse.kennesaw.edu/fye/submissionguidelines.php

Ensure you submit before the deadline listed on the lab schedule for CSE1322L here:
https://ccse.kennesaw.edu/fye/courseschedules.php

Rubric:

- Successfully copied and pasted the Encryption class (2 points)
- return_alphabet_index method (6 points total)
    - Takes in a character (2 points)
    - Finds the character in the array (2 points)
    - Returns the index (2 point)
- return_alphabet *method* (6 points total)
    - Takes in an integer (2 point)
    - Retrieves the character from the arraylist (2 point)
    - Returns the character (2 point)
- return_symbol_index method (6 points total)
    - Takes in a symbol (2 points)
    - Finds the symbol in the array (2 points)
    - Returns the index (2 point)
- return_symbol *method* (6 points total)
    - Takes in an integer (2 point)
    - Retrieves the symbol from the arraylist (2 point)
    - Returns the symbol (2 point)
- encrypt method (25 points total)
    - Pass in plain-text message via parameter (3 points)
    - Convert message to lower case (3 points)
    - Process the message one character at a time (19 points total)
        - Pull out one character from string (4 points)
        - Check for invalid character and return error message (4 points)
        - Encrypt character (4 points)
        - Concatenate new encrypted character to tmp string (4 points)
        - Return encrypted string (3 points)

- decrypt method (23 points total)
    - Pass in encrypted message via parameter (4 points)
    - Process the encrypted message one character at a time (19 points total)
        - Pull out one symbol from string (4 points)
        - Check for invalid symbol and return error message (4 points)
        - decrypt symbol (4 points)
        - Concatenate new decrypted character to tmp string (4 points)
        - Return decrypted string (3 points)
- Driver program (28 points total)
    - Print the menu (3 point)
    - Read the user choice (3 points)
    - If statement based on user input (3 points)
    - If they press 1 ask for plaintext, encrypt it and print result (3 points)
    - If they press 2 ask for ciphertext, decrypt it and print result (3 points)
    - If they press 3 terminate the program (3 points)
    - If they enter any other input, print error message and allow user to reenter (10 points)