|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor** |  | **Due Date** |  |

**PROJECT OOP in Java - Data Encryption and The Caesar Cipher 50 points**

**Objective** To use the Caesar cipher to encrypt login passwords.

***PROJECT DESCRIPTION***

Using the Caesar cipher, you will encrypt a set of passwords.

***Information about this Project***

A type of substitution cipher is the Caesar cipher. This cipher is mono - alphabetic since only one alphabet was used. In a substitution cipher, every letter of the plain text is substituted by the some other letter or symbol. The substitution in the Caesar cipher is based on a shift operation performed on the letters of the plaintext. For example if the shift 4 , then the standard alphabet

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |  |

becomes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |  |

and a plaintext message such as " PURCHASE NOW " is encrypted as " TYVGLEWI RSA " .

Notice that the letter P was shifted as T , the letter U as Y , and so on.

***Steps to Complete this Project***

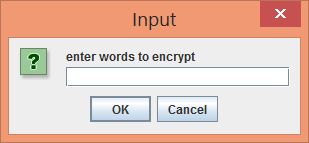
**STEP 1** **Open Eclipse, JCreator or Similar Java IDE**

Open Eclipse, JCreator or similar Java editor on your computer and create a new Java Project called Caesar. Next add a class to your project called Encryptor. Within your new class file, type the Java code from in **Figure 1** below.

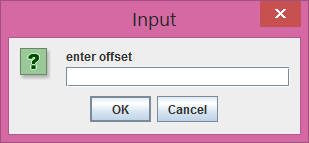
**STEP 2** **Test the File1**

Run your program. Your program should display an input box similar to the

screen snapshot segment given below.

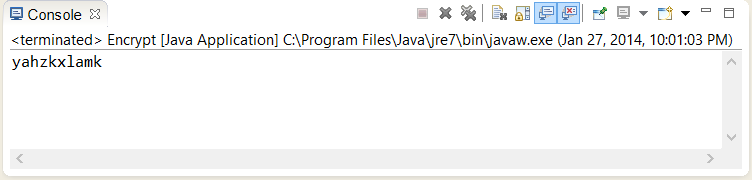


Into the text field, enter a password to encrypt, such as the word **subterfuge** and then in the next field **key**, enter a Caesar key shift such as the number **6** .



Observe the output that results.

**PROJECT OOP in Java - Data Encryption and The Caesar Cipher**



**STEP 3** **Dissect the Initial Program Code**

Examine in detail the code given in **Figure 1** . The bulk of the program code lies within the **Encryptor()** constructor that has a **String** return type.

**STEP 4** **Modify the File**

Return to your Java IDE and modify the file such that the user can enter a cipher text ( encrypted ) message and a key and your program will decrypt the message.

One way to modify the program is to first declare a class level variable.

**public String cipherText;**

Then supplement the **main()** method with the instantiation of a new object that will be used to perform the decryption.

Notice that in the main method the cipher text, i.e. the decrypted text, is assigned to a global variable in order for it to be used in the **Decrypt()** method.

|  |
| --- |
|  |

|  |
| --- |
| **public static void main(String [] args) {**    **// encryption block**  **Encryptor e = new Encryptor();**  **String strCipherText = e.Encrypt();**  **System.out.println(strCipherText);**    **// decrypt block**  **Encryptor d = new Encryptor();**  **// cipher text becomes the input text to the Decrypt method**  **d.cipherText = strCipherText;**  **String strPlainText = d.Decrypt();**  **System.out.println(strPlainText);**    **System.exit(0);**  **}** |

|  |
| --- |
|  |

**PROJECT OOP in Java - Data Encryption and The Caesar Cipher**

Within the class definition add a **Decrypt()** method. Copy the code from the **Encrypt()** method and place it into the **Decrypt()** method.

Now alter the copied code in the **Decrypt()** method as shown below such

|  |
| --- |
| **plainText = ((String)JOptionPane.*showInputDialog*("enter words " +**  **"to decrypt")).toLowerCase().trim();** |

to:

**plainText = cipherText;**

Finally change this copied block of code in the **Decrypt()** method shown below such that instead of performing a forward shift you will now perform a backward shift.

**offset += shift;**

**if(offset > 25)**

**{**

**newOffset = offset % 26;**

**sb.append(alphabet.charAt(newOffset));**

**}**

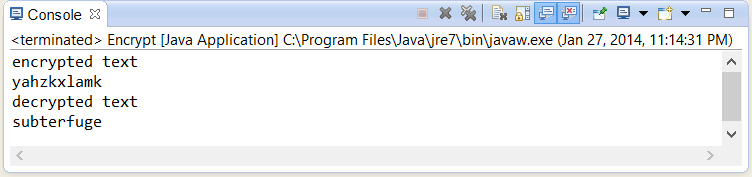
**else**

**{**

**sb.append(alphabet.charAt(offset) );**

**}**

A sample trial run of your modified program could appear as follows.



**STEP 5** **Submit the File**

Submit your original and modified source code file for credit. Include screen snapshots showing each the operation of your program with the above snapshotted encrypted and decrypted text results. Also zip up your entire

package and submit as well.

**PROJECT OOP in Java - Data Encryption and the Caesar Cipher**

**Figure 1 Caesar Cipher Program Code**

|  |
| --- |
|  |

|  |
| --- |
| **import javax.swing.\*;**  **public class Encryptor {**  **private String plainText;**  **private int shift;**    **public Encryptor() {**  **plainText = null;**  **shift = 0;**  **}**  **public static void main(String [] args) {**  **Encryptor e = new Encryptor();**  **String strCipherText = e.Encrypt();**  **System.out.println(strCipherText);**  **System.exit(0);**  **}**  **public String Encrypt()**  **{**  **plainText =**  **((String)JOptionPane.showInputDialog("enter words " +**  **"to encrypt")).toLowerCase().trim();**  **shift =**  **Integer.parseInt(JOptionPane.showInputDialog("enter offset"));**  **int offset = 0;**  **int newOffset = 0;**  **String alphabet = "abcdefghijklmnopqrstuvwxyz";**  **StringBuffer sb = new StringBuffer();**  **int index = plainText.length();**  **for(int i = 0; i < index; i++)**  **{**  **String temp = "" + plainText.charAt(i);**  **offset = alphabet.indexOf(temp);**  **offset += shift;**  **if(offset > 25)**  **{**  **newOffset = offset % 26;**  **sb.append(alphabet.charAt(newOffset));**  **}**  **else**  **{**  **sb.append(alphabet.charAt(offset));**  **}**  **}**  **return sb.toString();**  **}**  **}** |

|  |
| --- |
|  |