

PMR:

This program was easy with the help of the EIMACS module. I enjoyed this field of work and it was to experiment with the code. The code was kind of hard to build since there were so many lines to build though.

**ENCRYPTION**

import java.util.Scanner;

public class CeasarTester {

public static void menu() {

System.out.print("Would you like to (D)ecrypt or (E)ncrypt a Message or (Q)uit?: ");

Scanner in = new Scanner(System.in);

String choice = in.nextLine();

if (choice.equals("D")) {

System.out.print("Message to Decrypt: ");

String message = in.nextLine();

System.out.println();

System.out.print("Shift Used: ");

String shift = in.next();

int shiftInt = Integer.parseInt(shift);

CaesarShiftDecryption csd = new CaesarShiftDecryption(shiftInt, message);

csd.decryptMessage();

csd.printDecryptedMessage();

menu();

} else if (choice.equals("E")) {

System.out.print("Message to Encrypt: ");

String message = in.nextLine();

System.out.print("Shift to Use: ");

String shift = in.next();

int shiftInt = Integer.parseInt(shift);

CeasarShiftEncryption cse = new CeasarShiftEncryption(shiftInt, message);

cse.encryptMessage();

cse.printEncryptedMessage();

menu();

} else if (choice.equals("Q")) {

System.out.println("Goodbye.");

} else {

System.out.println("Error: Not a Valid Entry");

menu();

}

}

public static void setup() {

System.out.println("Caesar Shift Encryption/Decryption");

}

public static void main(String[] args) {

setup();

menu();

}

}

**DECRYPTION**

import java.util.ArrayList;

public class CaesarShiftDecryption {

private static int shift;

private static String message;

private static ArrayList<Character> Alphabet = new ArrayList<Character>();

private static ArrayList<Character> newMessage = new ArrayList<Character>();

public static void setupAlphabet() {

for (int i = 97; i < 123; i++) {

char letter = (char)i;

Alphabet.add(letter);

}

}

CaesarShiftDecryption(int s, String m) {

shift = s;

message = m;

setupAlphabet();

}

public static void decryptMessage() {

message = message.trim();

char[] msgFromUser = message.toCharArray();

for (int i = 0; i < msgFromUser.length; i++) {

if (msgFromUser[i] == ' ') {

newMessage.add(msgFromUser[i]);

} else {

int asciiToEncrypt = (int)msgFromUser[i];

asciiToEncrypt = asciiToEncrypt - shift;

if (asciiToEncrypt > 123) {

int overflow = asciiToEncrypt + 123;

int encryption = 97 - overflow;

char letter = (char)encryption;

newMessage.add(letter);

} else {

char letter = (char)asciiToEncrypt;

newMessage.add(letter);

}

}

}

}

public static int getShift() {

return shift;

}

public static void printDecryptedMessage() {

System.out.print("Encrypted Message: ");

for (int i = 0; i < newMessage.size(); i++) {

System.out.print(newMessage.get(i));

}

System.out.println("\n");

System.out.println("Shift Used: " + getShift());

}

}

**TESTER**

import java.util.Scanner;

public class CeasarTester {

public static void menu() {

System.out.print("Would you like to (D)ecrypt or (E)ncrypt a Message or (Q)uit?: ");

Scanner in = new Scanner(System.in);

String choice = in.nextLine();

if (choice.equals("D")) {

System.out.print("Message to Decrypt: ");

String message = in.nextLine();

System.out.println();

System.out.print("Shift Used: ");

String shift = in.next();

int shiftInt = Integer.parseInt(shift);

CaesarShiftDecryption csd = new CaesarShiftDecryption(shiftInt, message);

csd.decryptMessage();

csd.printDecryptedMessage();

menu();

} else if (choice.equals("E")) {

System.out.print("Message to Encrypt: ");

String message = in.nextLine();

System.out.print("Shift to Use: ");

String shift = in.next();

int shiftInt = Integer.parseInt(shift);

CeasarShiftEncryption cse = new CeasarShiftEncryption(shiftInt, message);

cse.encryptMessage();

cse.printEncryptedMessage();

menu();

} else if (choice.equals("Q")) {

System.out.println("Goodbye.");

} else {

System.out.println("Error: Not a Valid Entry");

menu();

}

}

public static void setup() {

System.out.println("Caesar Shift Encryption/Decryption");

}

public static void main(String[] args) {

setup();

menu();

}

}