```
//class to run and test program
import java.util.Scanner;
public class CaesarTester
{
   public static void main(String[] args)
   {
      //initializing new scanner
      Scanner in = new Scanner(System.in);

      String userChoice = "Y";

      Encryption tempE = new Encryption();
      Decryption tempD = new Decryption();

      while(userChoice.equalsIgnoreCase("Y"))
      {
         //asking for encryption or decryption
         System.out.println("Would you like to encode or decode a message? (E/D)");
         String cipherChoice = in.next();

         //making sure that input can be uppercase or lowercase
         if(cipherChoice.equalsIgnoreCase("E"))
         {
            //asking user to input number for key and checking for validity
            int shiftKey = -1;
            System.out.print("Please enter a value from 0-25 as a shift key for a Caesar Shift Cipher: ");
            shiftKey = in.nextInt();
            while(shiftKey <= -1 || shiftKey > 25)
            {
               System.out.println("The number entered was not within the specified domain. Please re-
enter a number between 0 and 25: ");
               shiftKey = in.nextInt();
            }
            System.out.println();

            System.out.println();

            //converting to cipher alphabet via shift skey
            String[] morphAlph = convertAlph(shiftKey);
            System.out.println();

            //displaying morphed alphabet
            System.out.print("Cipher Alphabet:   ");
            for(String letter: morphAlph)
            {
               System.out.print(letter + " ");
            }
            System.out.println();
```

```
            System.out.println();
            //prompting user to input message
            System.out.print("Please enter a message to encrypt:");
            String message1 = in.next();
            String message2 = in.nextLine();
            String message = message1 + message2;
            System.out.println();

            //calling encryption class to sucessfully encrypt a message
            String encryptedMessage = tempE.encrypt(message, morphAlph);
            System.out.println("Here is your encrypted message: \"" + encryptedMessage + "\" with shift
key " + shiftKey);
            System.out.println();
         }
         else
         {
             //asking user to input number for key and checking for validity
            int shiftKey = -1;
            System.out.print("Please enter a value from 0-25 as a shift key for a Caesar Shift Cipher: ");
            shiftKey = in.nextInt();
            while(shiftKey <= -1 || shiftKey > 25)
            {
               System.out.println("The number entered was not within the specified domain. Please re-
enter a number between 0 and 25: ");
               shiftKey = in.nextInt();
            }
            System.out.println();

            System.out.println();

            String[] morphAlph = convertAlph(shiftKey);
            System.out.println();

            //displaying morphed alphabet
            System.out.print("Cipher Alphabet:   ");
            for(String letter: morphAlph)
            {
               System.out.print(letter + " ");
            }
            System.out.println();
            System.out.println();
            //asking for user input for a message
            System.out.print("Please enter an encrpyped message to decrypt:");
            String message1 = in.next();
            String message2 = in.nextLine();
            String message = message1 + message2;
            System.out.println();
```

```
            //calling the decryption class to sucessfully decrypt a message given the shift key
            String decryptedMessage = tempD.decrypt(message, morphAlph);
            System.out.println("Here is your decrypted message: \"" + decryptedMessage + "\" with shift
key " + shiftKey);
            System.out.println();
         }

         //prompting user to continue using program
         System.out.print("Would you like to encode or decode another message? (Y/N) ");
         userChoice = in.next();
         System.out.println();
      }


   }

   public static String[] convertAlph(int key)
   {
      //shifting alphabet based on shift key to create new alphabet
      String[] morphAlph = new String[26];
      Encryption alph = new Encryption();
      String[] alphabet = alph.getALPHABET();
      int count = 0;
      for(int i = 0; i < morphAlph.length; i++)
      {
         if(i < morphAlph.length - key)
         {
            morphAlph[i] = alphabet[i+key];
         }
         else if(i >= morphAlph.length - key)
         {
            morphAlph[i] = alphabet[count];
            count++;
         }
      }

      //displaying original alphabet
      System.out.print("Original Alphabet: ");
      for(String letter: alphabet)
      {
         System.out.print(letter + " ");
      }

      return morphAlph;
   }
}
```

```
//class to encrypt messages
public class Encryption
{
   //constant for an alphabet
   private static final String[] ALPHABET = new
String[]{"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"
};

   public Encryption()
   {

   }

   public static String encrypt(String message, String[] ciphAlph)
   {
      String encryptMes = "";
      String alphabet = "";
      for(String character: ALPHABET)
      {
         alphabet += character;
      }

      //comparing letters input and transforming into the cipher alphabet based on shift key
      for(int i = 0; i < message.length(); i++)
      {
         String letter = message.substring(i,i+1);

         if(!letter.equals(" "))
         {
            int loc = alphabet.indexOf(letter);
            encryptMes += ciphAlph[loc];
         }
         else
         {
            encryptMes += " ";
         }

      }
      return encryptMes;
   }

   public String[] getALPHABET()
   {
      return ALPHABET;
   }
}

//class to decrypt messages
```

```
public class Decryption
{
   private static final String[] ALPHABET = new
String[]{"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"
};

   public Decryption()
   {

   }

   public static String decrypt(String message, String[] ciphAlph)
   {
      //accessing the alphabet list and comparing to the cipher alphabet
      String decryptMes = "";
      String ciphalpha = "";
      for(String character: ciphAlph)
      {
         ciphalpha += character;
      }

      for(int i = 0; i < message.length(); i++)
      {
         String letter = message.substring(i,i+1);

         //finding index of the cipher letter and transforming to normal alphabet
         if(!letter.equals(" "))
         {
            int loc = ciphalpha.indexOf(letter);
            decryptMes += ALPHABET[loc];
         }
         else
         {
            decryptMes += " ";
         }

      }
      return decryptMes;
   }

   //accessing the normal alphabet
   public String[] getALPHABET()
   {
      return ALPHABET;
   }
}
```