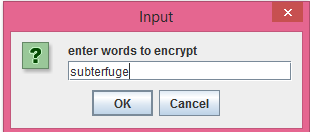
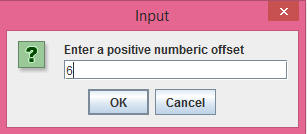
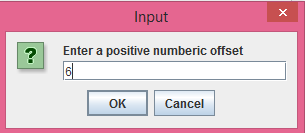
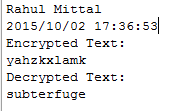
**Test Case 1: Positive Testing**

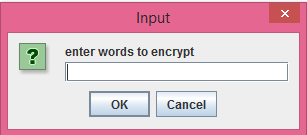


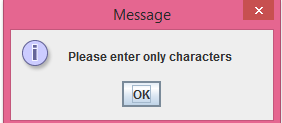


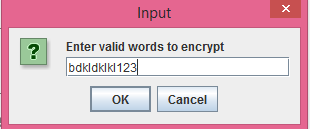


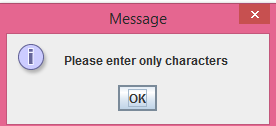


**Test Case 02: Negative Testing**

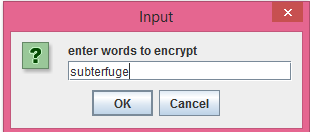


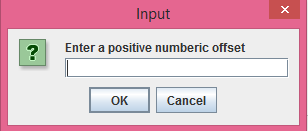




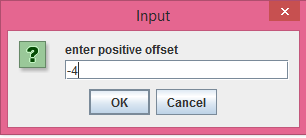


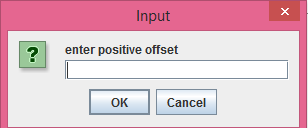
**Test Case 03: Negative testing for Offset Value:**

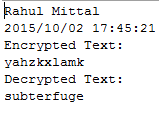












Code Snippet:

/\*\*

\* Name: Rahul Mittal

\* Lab Name: ITMD\_510\_LAB\_03

\* Current Date: 10/02/2015

\*/

**import** java.util.Date;

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** javax.swing.\*;

**public** **class** Encryptor

{

**private** String plainText;

**public** String cipherText;

**private** **int** shift;

/\*\*

\* Constructor to initialize the variables

\*/

**public** Encryptor()

{

plainText = **null**;

shift = 0;

}

/\*\*

\* Main method which will call all other methods

\* **@param** args

\*/

**public** **static** **void** main(String [] args)

{

DateFormat dateFormat = **new** SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

Date date = **new** Date();

Encryptor e = **new** Encryptor();

//This will get value from Encrypt Method

String strCipherText = e.Encrypt();

System.***out***.println("Rahul Mittal");

System.***out***.println(dateFormat.format(date)); //2015/09/07 14:19:25

System.***out***.println("Encrypted Text:");

System.***out***.println(strCipherText);

Encryptor d = **new** Encryptor();

d.cipherText = strCipherText;

//This will return decrypted text

String strPlainText = d.Decrypt();

System.***out***.println("Decrypted Text:");

System.***out***.println(strPlainText);

System.*exit*(0);

}

/\*\*

\* This function will get the offset value

\* and decrypt the encrytped value

\*/

**public** String Decrypt()

{

plainText = cipherText;

**try**

{

shift = Integer.*parseInt*(JOptionPane.*showInputDialog*("Enter a positive numberic offset"));

}

//If user has not entered any value, catch the NumberFormat Exception

**catch**(NumberFormatException nfe)

{

JOptionPane.*showMessageDialog*(**null**,"Do not left the value to blank");

System.*exit*(0);

}

**catch**(Exception e)

{

e.printStackTrace();

}

//Check if user has entered the negative value

//if yes, ask user to enter the positive value

**while**(shift < 0)

{

**try**

{

shift = Integer.*parseInt*(JOptionPane.*showInputDialog*("enter positive offset"));

}

**catch**(Exception e)

{

e.printStackTrace();

System.*exit*(0);

}

}

**int** offset = 0;

**int** newOffset = 0;

String alphabet = "abcdefghijklmnopqrstuvwxyz";

StringBuffer sb = **new** StringBuffer();

**int** index = plainText.length();

//Loop to convert encrypted value to decrypted one

//if the encryped value has suppose 'a' then the

//decrypted value should go back from z to a to check

//the value

**for**(**int** i = 0; i < index; i++)

{

String temp = "" + plainText.charAt(i);

offset = alphabet.indexOf(temp);

offset -= shift;

**if**(offset < 0)

{

newOffset = offset + 26;

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

}

**else**

{

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

}

}

**return** sb.toString();

}

/\*\*

\* This function will get the text value to be encypted

\* from the user and if the user has entered any special

\* character or a number, it will show a message that please

\* enter only numbers

\*/

**public** String Encrypt()

{

**try**

{

plainText =((String)JOptionPane.*showInputDialog*("enter words " +

"to encrypt")).toLowerCase().trim();

}

**catch**(Exception e)

{

e.printStackTrace();

}

//Check if user has entered any special character or number

**while**(!plainText.matches("[a-zA-Z]+"))

{

JOptionPane.*showMessageDialog*(**null**,"Please enter only characters");

plainText =((String)JOptionPane.*showInputDialog*("Enter valid words " +

"to encrypt")).toLowerCase().trim();

}

**try**

{

shift = Integer.*parseInt*(JOptionPane.*showInputDialog*("Enter a positive numberic offset"));

}

**catch**(NumberFormatException nfe)

{

JOptionPane.*showMessageDialog*(**null**,"Please enter valid positive offset");

System.*exit*(0);

}

**catch**(Exception e)

{

e.printStackTrace();

System.*exit*(0);

}

//Validate the offset that it should only be positive number

**while**(shift < 0)

{

**try**

{

shift = Integer.*parseInt*(JOptionPane.*showInputDialog*("enter positive offset"));

}

**catch**(Exception e)

{

e.printStackTrace();

System.*exit*(0);

}

}

**int** offset = 0;

**int** newOffset = 0;

String alphabet = "abcdefghijklmnopqrstuvwxyz";

StringBuffer sb = **new** StringBuffer();

**int** index = plainText.length();

//Loop to encrypt the text, it will check for

//each and every character and will encypt it

**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();

}

}