**Computer Science Project**

Encryptor - Decryptor

By Bino Oliver Manjesh,

Class XI A

# Table Of Contents

# 

[**Table Of Contents**](#_993ive7i5351) **2**

[**Introduction**](#_psybrkor10xj) **3**

[Class name: EncryptorDecryptor](#_f8n4unu0p21l) 3

[Methods:](#_xp4lduw4iczx) 3

[private String decrypt(String str, int key1, int key2)](#_r77y6qd0iqkt) 3

[private String encrypt(String str, int key1, int key2)](#_lq29ehix6z9z) 3

[private int getRandomKey()](#_e9lz6fm1o2iq) 3

[private static void printFormatted(String s)](#_9gput4rt9xn0) 3

[public static void main(String[] args)](#_rdsrzhdj7c9n) 3

[Working principle](#_7p9q4carahbo) 4

[Encryption:](#_t32asrfon3p3) 4

[Decryption:](#_y8i26cjoxwsc) 4

[**Source Code**](#_ld3vxcbijps4) **5**

**Output 8**

# 

# Introduction

Encryption is the process of encoding a message so that only authorised people can access it and decryption is the process of converting that encrypted message into the original message. This program accepts a sentence and encrypts or decrypts it. Before encryption, two encryption keys are generated which are used to encrypt the string. Decryption requires the two keys.

## Class name: EncryptorDecryptor

## Methods:

### **private** String decrypt(String str, **int** key1, **int** key2)

Decrypts the given string with the given keys.

### **private** String encrypt(String str, **int** key1, **int** key2)

Encrypts the given string with the given keys.

### **private int** getRandomKey()

Generates a random key between -256 and 256..

### **private static void** printFormatted(String s)

Prints a formatted message to the console. For example if “example” is passed to this method the output will be : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*example\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

Main method. Creates an object of Scanner class and EncryptorDecryptor and continuously accepts input until the user chooses to exit.

## Working principle

### 

### Encryption:

Each character is individually changed into a different one. The unicode of the new character is obtained by the following calculation:-

Where i = index of the character.

Note: As the position also plays a role in the transformation, the same character may have a different character in the encrypted string if it is in a different position.

### Decryption:

Each character is decrypted individually. The unicode for the character is obtained by reversing the above calculation.

# 

# Source Code

**import** java.util.InputMismatchException;

**import** java.util.Scanner;

**class** EncryptorDecryptor {

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

Scanner sc = **new** Scanner(System.***in***);

EncryptorDecryptor obj = **new** EncryptorDecryptor();

*printFormatted*(**""**);

*printFormatted*(**"ENCRYPTOR/DECRYPTOR"**);

*printFormatted*(**""**);

**boolean** end = **false**;

**while** (!end) {

System.***out***.println(**"ENTER:-"**);

System.***out***.println(**"1: TO ENCRYPT A STRING"**);

System.***out***.println(**"2: TO DECRYPT A STRING"**);

System.***out***.println(**"0: TO EXIT"**);

**try** {

**int** ch = sc.nextInt();

sc.nextLine();

String str, encrypted;

**int** key1, key2;

**switch** (ch) {

**case** 1:

System.***out***.println(**"ENTER THE STRING"**);

str = sc.nextLine();

key1 = obj.getRandomKey();

key2 = obj.getRandomKey();

encrypted = obj.encrypt(str, key1, key2);

System.***out***.println(**"THE ENCRYPTED TEXT:-"**);

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

System.***out***.println(**"KEY 1: "** + key1);

System.***out***.println(**"KEY 2: "** + key2);

**break**;

**case** 2:

System.***out***.println(**"ENTER THE STRING"**);

encrypted = sc.nextLine();

System.***out***.print(**"ENTER KEY 1: "**);

key1 = sc.nextInt();

System.***out***.print(**"ENTER KEY 2: "**);

key2 = sc.nextInt();

str = obj.decrypt(encrypted, key1, key2);

System.***out***.println(**"THE DECRYPTED TEXT:-"**);

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

**break**;

**case** 0:

end = **true**;

*printFormatted*(**""**);

*printFormatted*(**"THANK YOU FOR USING THIS PROGRAM"**);

**break**;

**default**:

System.***out***.println(**"PLEASE ENTER ONLY 0, 1 OR 2"**);

}

} **catch** (InputMismatchException e) {

System.***out***.println(**"YOU MAY HAVE ENTERED A STRING WHERE A NUMBER WAS EXPECTED"**);

sc = **new** Scanner(System.***in***);

} **catch** (Exception e) {

System.***out***.println(**"AN UNKNOWN ERROR HAS OCCURED"**);

}

*printFormatted*(**""**);

}

}

**private static void** printFormatted(String s) {

**int** padLength = (50 - s.length()) / 2;

**for** (**int** i = 0; i < padLength; i++) {

System.***out***.print(**"\*"**);

}

System.***out***.print(s);

padLength = 50 - (padLength + s.length());

**for** (**int** i = 0; i < padLength; i++) {

System.***out***.print(**"\*"**);

}

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

}

**private int** getRandomKey() {

**return** (**int**) (Math.*random*() \* 512) - 256;

}

**private** String encrypt(String str, **int** key1, **int** key2) {

StringBuffer encrypted = **new** StringBuffer();

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

**char** c = str.charAt(i);

**char** newC = (**char**) (c \* c + key1 + i \* key2);

encrypted.append(newC);

}

**return** encrypted.toString();

}

**private** String decrypt(String str, **int** key1, **int** key2) {

StringBuffer decrypted = **new** StringBuffer();

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

**char** c = str.charAt(i);

**char** oldC = (**char**) ((**int**) Math.*sqrt*(c - key1 - i \* key2));

decrypted.append(oldC);

}

**return** decrypted.toString();

}

}

Output

