

Autocipher.java

```
1/*This program enables us to encrypt plain text to a cipher text using Auto Key Cipher
2 */
3
4/*imports the scanner class from the
5 *util package used for reading files.*/
6import java.util.Scanner;
7
8/*imports the regular expression class from
9 * util package to enable the program to
10 * understand regular expressions.*/
11import java.util.regex.Pattern;
12
13//imports the package I/O
14import java.io.*;
15
16/*imports nio package with File and Path classes enabling writing to a
17 *file directly without use of traditional I/O operations.
18 *Supported by JDK-7 and up*/
19import java.nio.file.Files;
20import java.nio.file.Paths;
21/**
22 *
23 */
24
25/**
26 * @author Sunit Tiwari
27 *
28 */
29public class Autocipher {
30    // Creating an object for Scanner class and using it to reference it whenever needed.
31    private static Scanner scanner = new Scanner( System.in );
32
33    // Creating an object for Autocipher class and using it for method call.
34    private static Autocipher methodCall = new Autocipher();
35
36    /**// Initializes the entire alphabets in a string which will be referenced upon
37     * as index value to compute the required letter. */
38    public String autokeyval = "ABCDEFGHGIJKLMNOPQRSTUVWXYZ";
39    //Global variables
40    public String newKey = new String();
41    public String decrypt = new String();
42    public String temp = new String();
43    public String key = new String();
44    public String ciphertext = new String();
45
46
47    /**
48     * @param args
49     * @throws IOException
50     The main function call */
51    public static void main(String[] args) throws IOException {
52
53        System.out.println("Enter your name");
54        String input = scanner.nextLine(); //Stores the input provided by the user in a string
55        with the help of scanner class
56
57        System.out.println("Welcome" +"@" +input+" "+" Choose any one from the list below");
```

Autocipher.java

```

57     System.out.println("1.ENCRYPTION");
58     System.out.println("2.EXIT");
59
60     int option = scanner.nextInt();
61     if (option == 1) {
62         methodCall.encryption(); // Encryption method being called.Jumps the execution to
        encryption method
63         System.out.println("ENCRYPTION Done.The encrypted File is saved as encrypt.txt and
        the key has been stored in Keyword.txt");
64
65     } else if (option == 2) {
66         System.out.println("See you later"+" "+input+"!!!! Bye Bye :)");
67         System.exit(0);
68     }
69
70 }
71
72 public void encryption()
73 {
74     try {
75         /*To implement the Auto cipher we need a key.
76         *Here we are getting the key as an Input from the user.
77         *The key is a single word which should not have any spaces,special
        characters,punctuation eg:- welcome*/
78         Scanner auto_key = new Scanner(System.in);
79         System.out.println("Please enter a single word without space and any special
        character");
80         key = auto_key.next();
81         Pattern pattern = Pattern.compile("[A-Za-z]++"); //Using regular expression to
        validate the input provided by the user.
82         if (!pattern.matcher(key).matches()) {
83             throw new IllegalArgumentException("Invalid String");
84         }
85         else
86         {
87             /*Now we need to check whether the input file size is greater>1 mb.
88             *If greater then discard the file and throw exception.*/
89             File file = new File ("./input.txt");
90             long filesizeinBytes = file.length();
91             long filesizeinKB = (filesizeinBytes/1024);
92             long filesizeinMB = (filesizeinKB/1024);
93             if (filesizeinMB >1) {
94                 throw new IllegalArgumentException("File Size greater than 1 MB");
95             }
96             /*If the file size is less than 1 mb and it is not empty then
97             *pull out the data into a string from the text file.*/
98             System.out.println("File size is less than 1 mb");
99             BufferedReader bufferedReader = new BufferedReader(new
        FileReader("input.txt"));
100             StringBuffer stringBuffer = new StringBuffer();
101             String line = null;
102             while((line =bufferedReader.readLine())!=null){
103
104                 stringBuffer.append(line).append("\n");
105                 /*Now we format the string.Remove the spaces,
106                 *punctuation and special character if any*/
107                 String withoutspace = line.replaceAll("\\s", "");

```

Autocipher.java

```

108         System.out.println(withoutspace);
109         String withoutspecialchar = withoutspace.replaceAll("\\W", "");
110         System.out.println(withoutspecialchar);
111         String withoutnumber = withoutspecialchar.replaceAll("\\d", "");
112         temp = withoutnumber.toUpperCase(); // Convert the entire string to
uppercase to remove ambiguity.
113         System.out.println(temp);
114
115     }
116     String simple = ((key+temp)).toUpperCase();
117     /*Writing the key to a text file.
118     *RandomAccessFile allows us to replace the old key with the latest one each
time the program is run*/
119     RandomAccessFile f = new RandomAccessFile(new File("keyword.txt"), "rw");
120     f.seek(0);
121     f.write(key.getBytes());
122     f.close();
123     /*Below we generate subkey of equal length as the plain text.
124     *And generate cipher text using the auto key cipher*/
125     for (int i = 0; i < temp.length(); i++) {
126         char subkey = simple.charAt(i);
127         newKey += Character.toString(subkey);
128     }
129     System.out.println(newKey);
130     for (int index = 0; index < temp.length(); index++) {
131         int inputFileTextVal = autokeyval.indexOf(temp.charAt(index));
132         int newkeyVal = autokeyval.indexOf(newKey.charAt(index));
133         int cipherVal = (inputFileTextVal+newkeyVal)%26;
134         System.out.println(inputFileTextVal);
135         System.out.println(newkeyVal);
136         System.out.println(cipherVal);
137
138         ciphertext += autokeyval.charAt(cipherVal);
139         System.out.println(ciphertext);
140         Files.write(Paths.get("./encrypt.txt"), ciphertext.getBytes());} // Writes
the cipher text to a file
141
142     }
143     } catch (FileNotFoundException e) {
144
145         e.printStackTrace();
146     }
147
148     catch (IOException e) {
149
150         e.printStackTrace();
151     }
152 }
153
154 }
155
156
157
158

```