

**A REPORT  
ON  
PASSWORD GENERATOR  
AND  
STRENGTH CHECKER**

**By  
Shefali Khera(22csu426)  
Mahith Rajeev(22csu110)  
Manan Mittal(22csu365)  
Manik Sehrawat(22csu112)**

**23/05/2023**

**To  
Ms. Akanksha Kaushik  
North Cap University**

## **ACKNOWLEDGEMENT**

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our highly respected and esteemed guide, Ms. Akanksha Kaushik for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behaviour are sincerely acknowledged. We also wish to express my indebtedness to all the faculty members of the department for their cooperation and suggestions.

Place: NCU,

Haryana

Date: 23/05/2023

## **TABLE OF CONTENTS**

<b>SNO</b>	<b>CONTENTS</b>	<b>PAGE NO.</b>
1	Problem statement	4
2	Introduction	4
3	Why to use password generator	5
4	Algorithm	6
5	Flowchart	7
6	Class Diagram	8-9
7	Advantages	10
8	Code and output	11-12
9	Limitations	13
10	References	13

## **PROBLEM STATEMENT:**

A java application to generate passwords based on user requirement and calculate the strength of the password chosen by the user to maintain high security and less risk of account hacking.

## **INTRODUCTION:**

The password generator and strength checker project are a software application that creates and evaluates strong and unique passwords for online accounts. It utilizes Java programming language to provide high security and resilience against potential cyber threats. With a user-friendly interface, the project promotes better password practices by recommending ways to improve existing passwords and generating new ones that are difficult to guess or crack. The project aims to enhance online security and protect sensitive data from unauthorized access, contributing to a safer and more secure digital environment.

- The code takes user input for a password and generates 3 passwords and let the user choose it, then performs several checks on the password to determine its strength, such as checking if it contains at least one uppercase letter, lowercase letter, number, and special symbol.
- Based on the number of checks passed, the code assigns a strength level to the password and returns a message to the user indicating the strength level. The possible strength levels are weak, good and strong.
- The code uses a static method named generate Password to perform the password checks and return the strength level message. It then calls this method from the main method, which takes user input and outputs the generated message to the console.
- A password strength tester gauges how long it might hypothetically take to crack your password by testing the password against a set of known criteria—such as length, randomness, and complexity.

## **WHY TO USE PASSWORD GENERATOR?**

Password generators are valuable for several reasons.

- They allow you to create long, complex passwords. Password generators are also beneficial if you have trouble coming up with passwords that meet specific requirements (e.g., at least 12 characters, numbers, symbols, etc.).
- They allow you to create different passwords for different websites so that if one gets compromised the others remain safe. This can help prevent hackers from breaching one system and getting access to all your accounts.
- Most importantly, password generators are widely available across different computing platforms. They can be accessed from websites, desktops, laptops, tablets, smartphones, and other mobile devices.
- This is particularly important if you often access the internet using public computers. When this is the case, you're often forced to create weak passwords or risk getting logged out by the system administrator

## **ALGORITHM:**

The algorithm of the provided code is as follows:

**STEP 1:** The main method is the entry point of the program. It prompts the user to specify the password requirements such as including uppercase letters, lowercase letters, numbers, symbols, and the desired length of the password.

**STEP 2:** The user's input is read using a Scanner object.

**STEP 3:** An array passwordArray is initialized to store generated passwords, and another array passwordstatus is initialized to store the status of each password (weak, medium, good, or strong).

**STEP 4:** A loop is executed three times to generate three passwords based on the user's requirements. Inside the loop, the generatePassword method is called from the PasswordGenerator class, passing the user's input as arguments. The generated password is stored in the passwordArray, and its status is determined using the setStatus method.

**STEP 5:** The generated password and its status are printed to the console.

**STEP 6:** The user is prompted to choose one password from the generated ones to add to a file. The selected password is passed to the writepasswordtofile method along with the filename provided by the user.

**STEP 7:** The setStatus method calculates the score of a given password based on its length and various criteria (presence of uppercase letters, lowercase letters, numbers, and symbols). The score is used to determine the password's status (weak, medium, good, or strong).

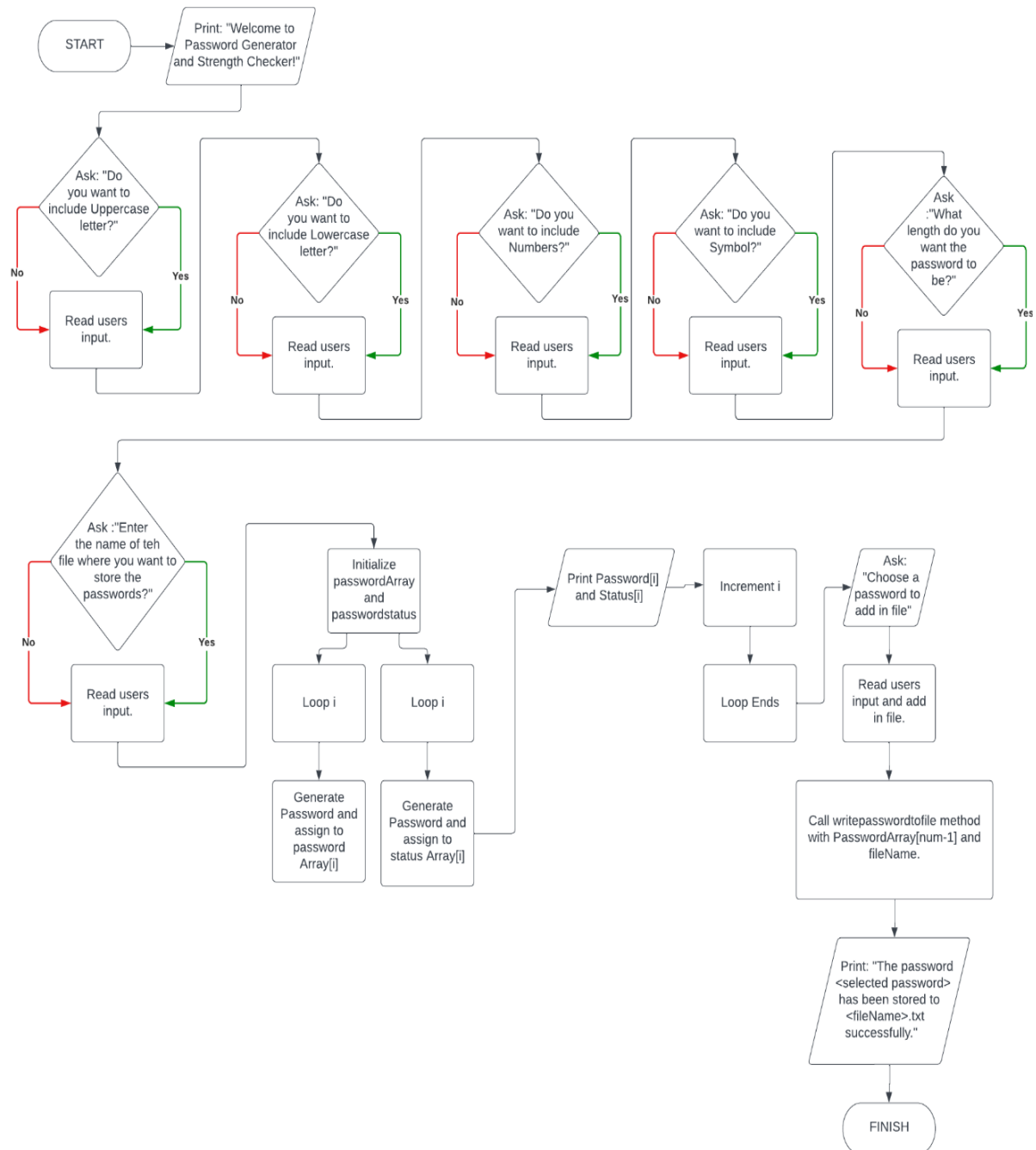
**STEP 8:** The writepasswordtofile method takes a password and a filename as parameters and writes the password to the specified file using a FileWriter object.

**STEP 9:** The PasswordGenerator class contains two methods:

The generatePassword method takes the user's requirements and the desired length as parameters. It generates a password by randomly selecting characters from the available character sets (uppercase letters, lowercase letters, numbers, symbols) based on the user's choices.

The calculatePasswordScore method takes a password as a parameter and calculates its score based on length and various criteria. The score is used to measure the password's strength.

## FLOWCHART:



## CLASS DIAGRAM:

### Generatepassword()

```
String      uppercaseLetters    =  
"ABCDEFGHJKLMNOPQRSTUVWXYZ";  
  
String      lowercaseLetters    =  
"abcdefghijklmnopqrstuvwxyz";  
  
String numbers = "0123456789";  
  
String      symbols      =      "!@#%&*()_+  
=[]{\\}|;:'\"./<>?";
```

### Writepasswordtofile()

```
Filewriter();
```

```
try {  
    FileWriter      writer      =      new  
    FileWriter(fileName);  
    writer.write("Password : "+pswd);  
    writer.close();  
    System.out.println("The password :  
"+pswd+" has been stored to  
"+fileName+".txt successfully.");  
} catch (IOException e) {  
    System.out.println("Error      writing  
passwords to file.");  
}
```

### calculatepasswardscore()

```
int score = 0;  
  
if (password.length() >= 8) {  
    score += 10;  
}  
  
if (password.matches("(?=.*\\d).*")) {  
    score +  
  
if (password.matches("(?=.*[a-z]).*")) {  
    score += 10;  
}  
  
if (password.matches("(?=.*[A-Z]).*")) {  
    score += 10;  
}  
  
= 10;  
}  
  
if      (password.matches("(?=.*[!@#%&*()_+  
=\\[\\]{}\\\\\\|;:'\"./<>?]).*")) {  
    score += 10;
```



## Pswd()

```
input; useUppercase; useLowercase;  
useNumbers  
useSymbols;password  
writepasswordtofile(passwordArray[num-  
1],fileName);
```

## setStatus()

```
int num;  
String status;
```

## **ADVANTAGES:**

There are many advantages to using a password generator. Among them are the following:

- Easy creation of strong passwords.
- Allows the use of different passwords for each website.
- Accessible from any computer or other device
- Can be used with any type of account – email accounts, social media sites, banking portals, etc.
- Can be used offline.
- No need to remember passwords.
- Protects accounts in case of hardware failure, natural disasters, or service shutdown – If you store all your passwords in one place and that device fails, you risk getting locked out of all your accounts.

## CODE:

```
1  import java.io.*;
2  import java.util.*;
3
4
5  public class Pwd extends PasswordGenerator {
6      public static void main(String[] args) {
7          Scanner input = new Scanner(System.in);
8          System.out.println("Welcome to Password Generator and Strength Checker!");
9          System.out.println("Do you want to include uppercase letters? (yes/no)");
10         String useUpperCase = input.nextLine().toLowerCase();
11         System.out.println("Do you want to include lowercase letters? (yes/no)");
12         String useLowerCase = input.nextLine().toLowerCase();
13         System.out.println("Do you want to include numbers? (yes/no)");
14         String useNumbers = input.nextLine().toLowerCase();
15         System.out.println("Do you want to include symbols? (yes/no)");
16         String useSymbols = input.nextLine().toLowerCase();
17         System.out.println("What length do you want the password to be?");
18         int length = input.nextInt();
19         System.out.println("Enter the name of the file where you want to store the passwords:");
20         String fileName = input.next();
21         String[] passwordArray = new String[3];
22         String[] passwordstatus = new String[3];
23         for (int i = 0; i < 3; i++) {
24             String password = generatePassword(useUpperCase, useLowerCase, useNumbers, useSymbols, length);
25             passwordArray[i] = password;
26             passwordstatus[i] = setStatus(passwordArray[i]);
27             System.out.println("Password " + (i + 1) + ": " + password);
28             System.out.println("Status : " + passwordstatus[i]);
29         }
30         System.out.println("Choose one password to add in file");
31         int num = input.nextInt();
32         writepasswordtofile(passwordArray[num - 1], fileName);
33     }
34 }
```

[Finished in 1.0s]

```
34
35     public static String setStatus(String pswd) {
36         String status;
37         int num = calculatePasswordScore(pswd);
38         if (num < 25) {
39             status = "Weak";
40         } else if (num < 50) {
41             status = "Medium";
42         } else if (num < 75) {
43             status = "Good";
44         } else {
45             status = "Strong";
46         }
47         return status;
48     }
49
50     public static void writepasswordtofile(String pswd, String fileName) {
51         try {
52             FileWriter writer = new FileWriter(fileName);
53             writer.write("Password : " + pswd);
54             writer.close();
55             System.out.println("The password : " + pswd + " has been stored to " + fileName + ".txt successfully.");
56         } catch (IOException e) {
57             System.out.println("Error writing passwords to file.");
58         }
59     }
60 }
61
62 class PasswordGenerator {
63     public static String generatePassword(String useUpperCase, String useLowerCase, String useNumbers, String useSymbols, int
64     String uppercaseLetters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
65     String lowercaseLetters = "abcdefghijklmnopqrstuvwxyz";
66     String numbers = "0123456789";
67     String symbols = "!@#$%^&*()_+-=[]{}|\\|';:~.,/<>?";
68     String availableChars = "";
69     if (useUpperCase.equals("yes")) {
70         availableChars += uppercaseLetters;
71     }
72     if (useLowerCase.equals("yes")) {
73         availableChars += lowercaseLetters;
74     }
75     if (useNumbers.equals("yes")) {
76         availableChars += numbers;
77     }
78     if (useSymbols.equals("yes")) {
79         availableChars += symbols;
80     }
81     Random random = new Random();
82     String password = "";
83     for (int i = 0; i < length; i++) {
84         int randomIndex = random.nextInt(availableChars.length());
85         char randomChar = availableChars.charAt(randomIndex);
86         password += randomChar;
87     }
88     return password;
89 }
90
91 public static int calculatePasswordScore(String password) {
92     int score = 0;
93     if (password.length() >= 8) {
```

[Finished in 1.0s]

```
62 class PasswordGenerator {
63     public static String generatePassword(String useUpperCase, String useLowerCase, String useNumbers, String useSymbols, int
64     String uppercaseLetters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
65     String lowercaseLetters = "abcdefghijklmnopqrstuvwxyz";
66     String numbers = "0123456789";
67     String symbols = "!@#$%^&*()_+-=[]{}|\\|';:~.,/<>?";
68     String availableChars = "";
69     if (useUpperCase.equals("yes")) {
70         availableChars += uppercaseLetters;
71     }
72     if (useLowerCase.equals("yes")) {
73         availableChars += lowercaseLetters;
74     }
75     if (useNumbers.equals("yes")) {
76         availableChars += numbers;
77     }
78     if (useSymbols.equals("yes")) {
79         availableChars += symbols;
80     }
81     Random random = new Random();
82     String password = "";
83     for (int i = 0; i < length; i++) {
84         int randomIndex = random.nextInt(availableChars.length());
85         char randomChar = availableChars.charAt(randomIndex);
86         password += randomChar;
87     }
88     return password;
89 }
90
91 public static int calculatePasswordScore(String password) {
92     int score = 0;
93     if (password.length() >= 8) {
```

[Finished in 1.0s]

## OUTPUTS:

```
C:\Windows\System32\cmd.exe X + v
Welcome to Password Generator and Strength Checker!
Do you want to include uppercase letters? (yes/no)
yes
Do you want to include lowercase letters? (yes/no)
yes
Do you want to include numbers? (yes/no)
yes
Do you want to include symbols? (yes/no)
yes
What length do you want the password to be?
9
Enter the name of the file where you want to store the passwords:
Shefali
Password 1: [B%vZ+aG8
Status : Strong
Password 2: x>$p\BfLC
Status : Good
Password 3: f-(PZ@Bmx
Status : Good
Choose one password to add in file
1
The password : [B%vZ+aG8 has been stored to Shefali.txt successfully.
C:\Users\Dell\OneDrive\Documents>

Finished in 1.2s]
```

```
C:\Windows\System32\cmd.exe X + v
Welcome to Password Generator and Strength Checker!
Do you want to include uppercase letters? (yes/no)
yes
Do you want to include lowercase letters? (yes/no)
yes
Do you want to include numbers? (yes/no)
yes
Do you want to include symbols? (yes/no)
yes
What length do you want the password to be?
9
Enter the name of the file where you want to store the passwords:
Shefali
Password 1: BtdtJ"=x>
Status : Good
Password 2: vRJ}.N]4f
Status : Strong
Password 3: fUN[q<oQ?
Status : Good
Choose one password to add in file
3
Error writing passwords to file.
C:\Users\Dell\OneDrive\Documents>
```

## **LIMITATIONS:**

- they create long strings of random characters that are highly difficult for humans to remember.
- While password generators are a great tool for improving password management, they aren't a substitute for other security measures.
- It's critical to take steps to protect your devices from malware and viruses, install automatic updates, and set up two-step verification. This will help protect your devices, data, and accounts from hackers and malicious software.

## **REFERENCES:**

<https://www.kaspersky.co.in/resource-center/threats/how-to-create-a-strong-password>

<https://iq.opengenus.org/password-generator-and-strength-checker-in-java/>

<https://www.javatpoint.com/java-password-generator>

# THANK YOU!