AQA
Realising potential

GCSE COMPUTER SCIENCE
(8520)

June 2017/18 NEA Task
Password checker and generator

Example solution 2

Version 1.0 June 2018

NEA EXAMPLE SOLUTION

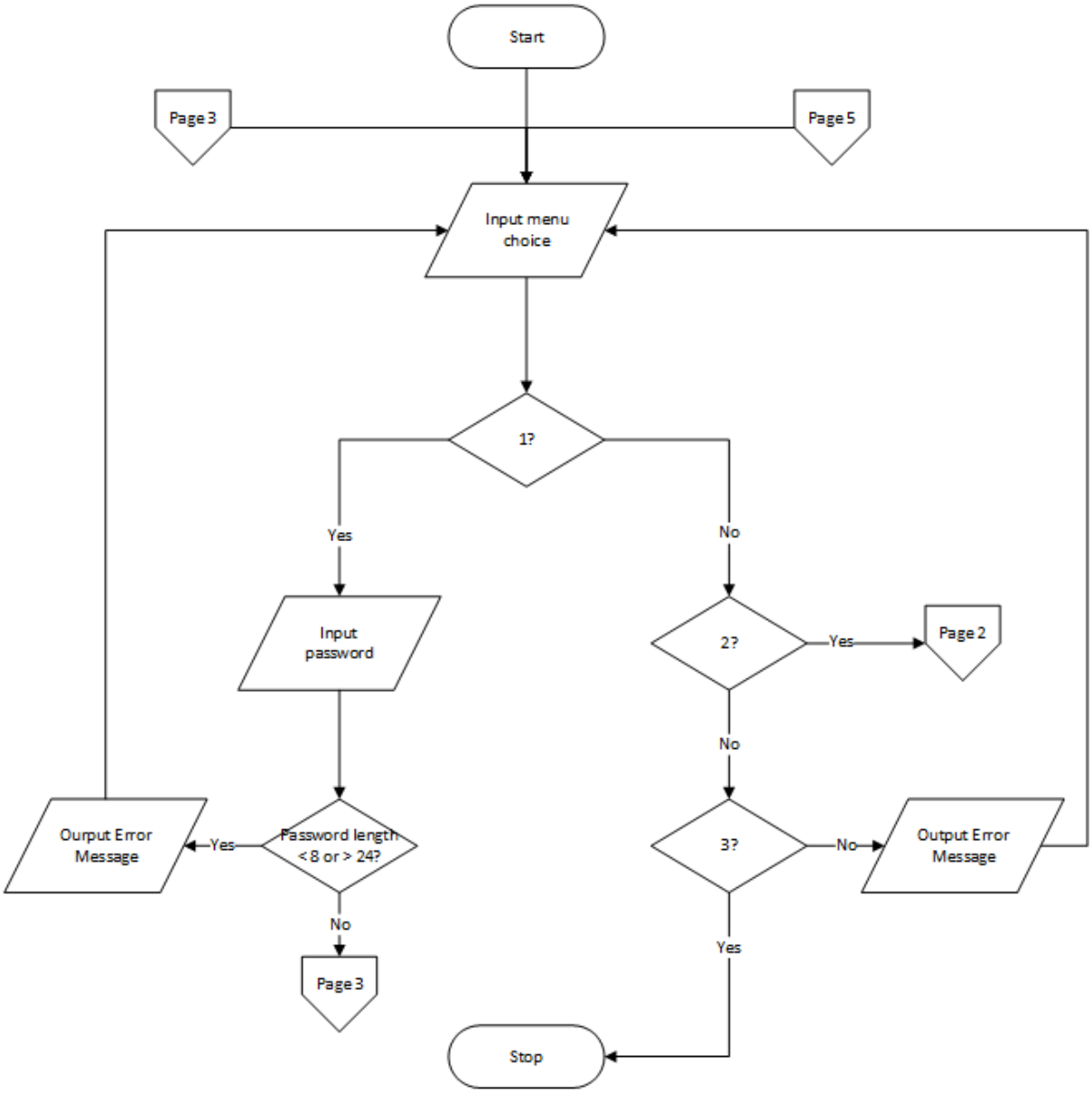
Introduction

The attached example scenario solution is provided to give teachers an indication of the type of solution that students could complete in response to the June 2017/18 NEA scenario: Task 1 – Password checker for the new GCSE Computer Science (8520) specification.

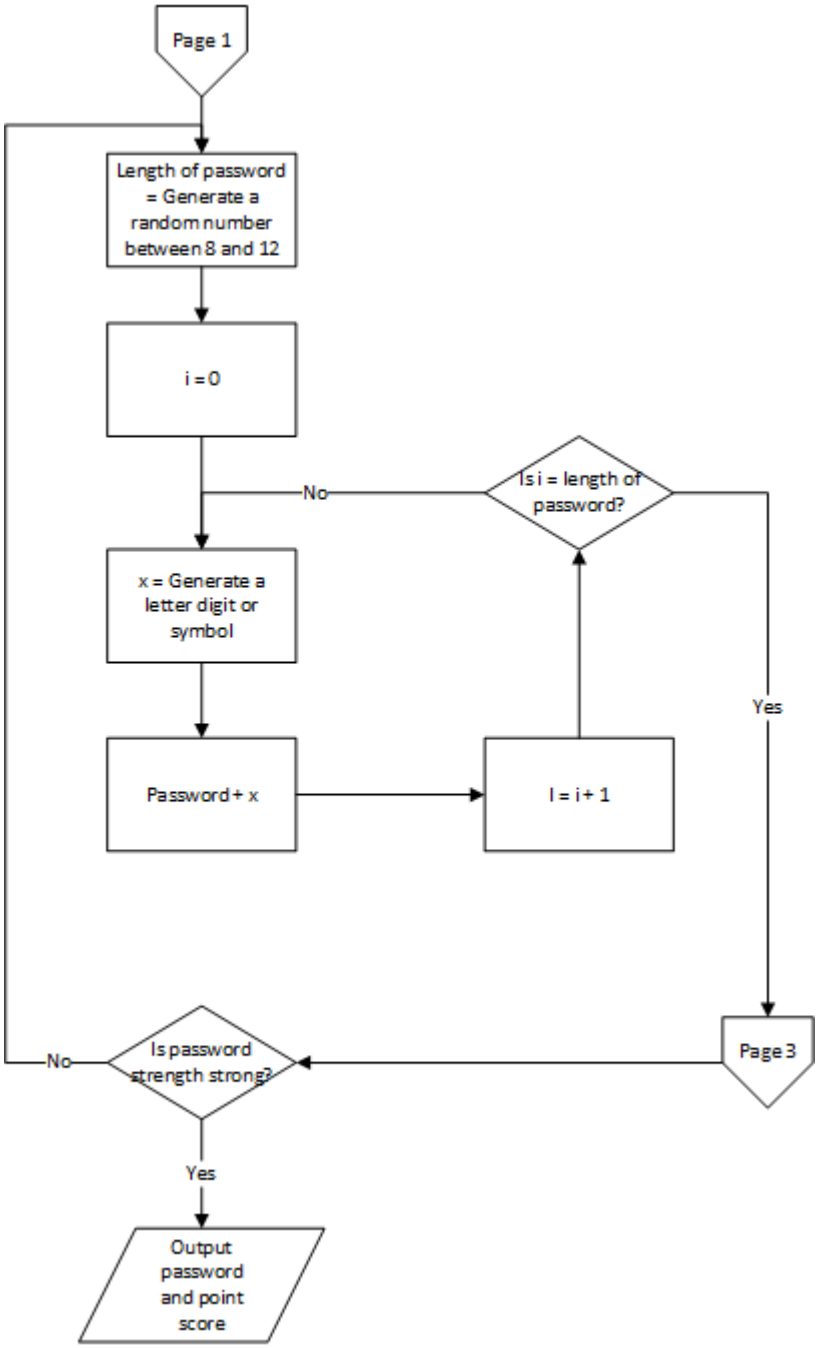
This example solution should only be used to enable teachers to commence planning work for the NEA, (the live NEA scenario will be available from September 2018). As a result teachers should use this only as a guide to the forthcoming live scenarios. The solution is not a ‘real’ solution and is provided as an example only.

Design

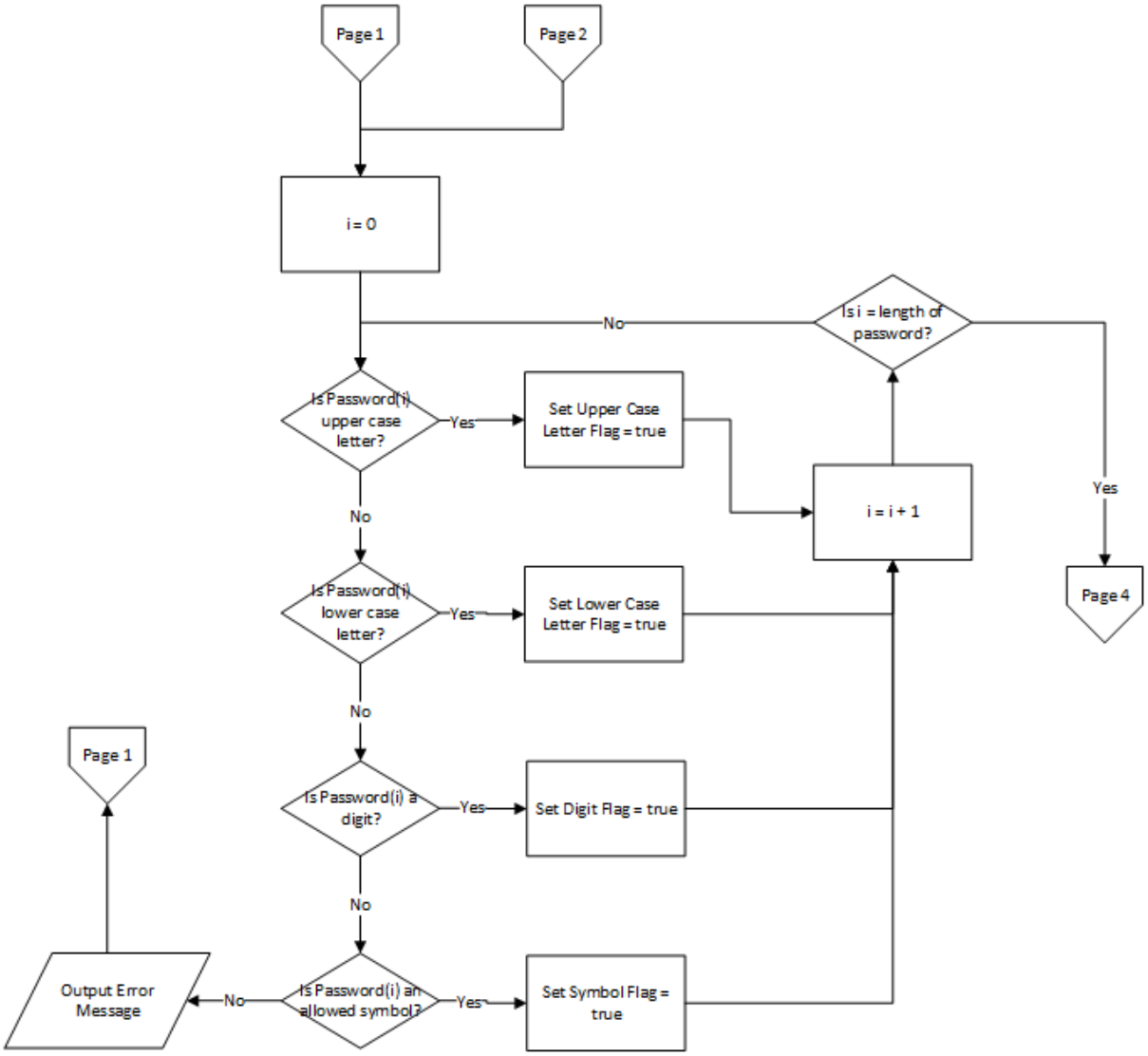
Page 1



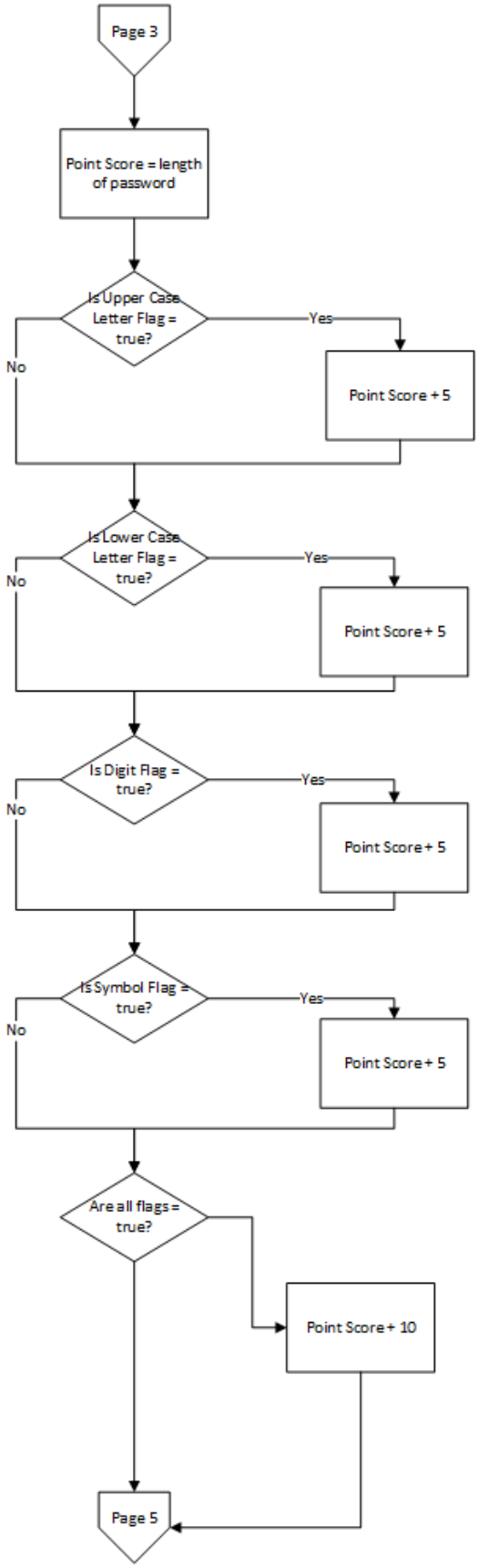
Page 2


Page 1
Length of password = Generate a random number between 8 and 12
i = 0
Is i = length of password?
No
x = Generate a letter digit or symbol
Password + x
I = i + 1
Yes
Page 3
Is password strength strong?
No
Yes
Output password and point score

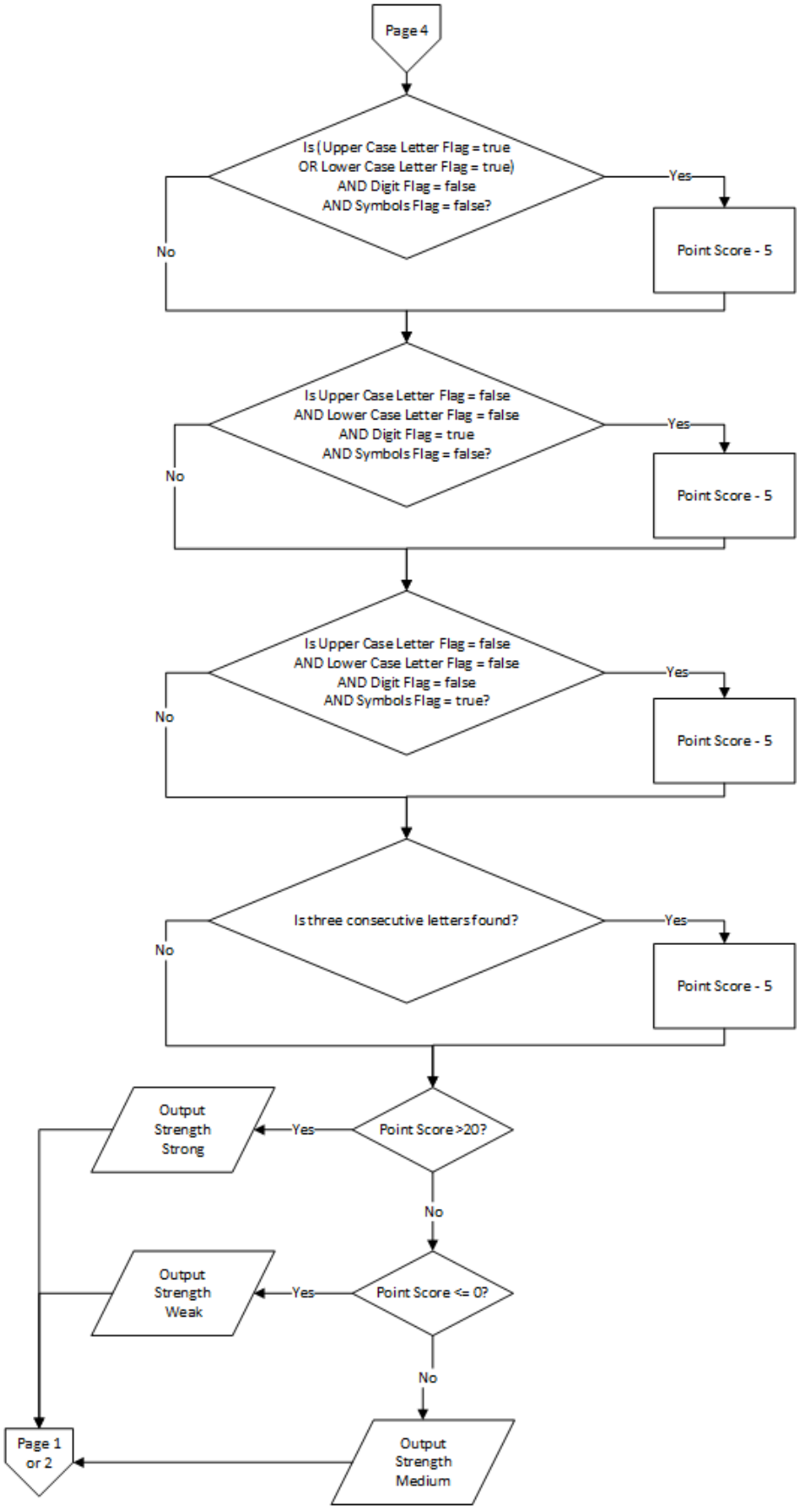
Page 3


Page 1
Page 2
i = 0
Is i = length of password?
No
Yes
Page 4
Is Password(i) upper case letter?
Yes
Set Upper Case Letter Flag = true
No
i = i + 1
Is Password(i) lower case letter?
Yes
Set Lower Case Letter Flag = true
No
Page 1
Is Password(i) a digit?
Yes
Set Digit Flag = true
No
Output Error Message
No
Is Password(i) an allowed symbol?
Yes
Set Symbol Flag = true

Page 4


Page 3
Point Score = length of password
Is Upper Case Letter Flag = true?
Yes
No
Point Score + 5
Is Lower Case Letter Flag = true?
Yes
No
Point Score + 5
Is Digit Flag = true?
Yes
No
Point Score + 5
Is Symbol Flag = true?
Yes
No
Point Score + 5
Are all flags = true?
Point Score + 10
Page 5

Page 5


Page 4
Is (Upper Case Letter Flag = true
OR Lower Case Letter Flag = true)
AND Digit Flag = false
AND Symbols Flag = false?
Yes
No
Point Score - 5
Is Upper Case Letter Flag = false
AND Lower Case Letter Flag = false
AND Digit Flag = true
AND Symbols Flag = false?
Yes
No
Point Score - 5
Is Upper Case Letter Flag = false
AND Lower Case Letter Flag = false
AND Digit Flag = false
AND Symbols Flag = true?
Yes
No
Point Score - 5
Is three consecutive letters found?
Yes
No
Point Score - 5
Point Score >20?
Yes
Output Strength Strong
No
Point Score <= 0?
Yes
Output Strength Weak
No
Output Strength Medium
Page 1 or 2

Code

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Scanner;

public class NEA {
    public static Scanner scan = new Scanner(System.in);
    // This is used to scan in anything entered by the user. A public scanner
    // because local scanners can't be closed and then called again later.

    public static void main(String args[]) {
        menu();
    }

    public static void menu() {
        // This displays the menu and tells the program which one of three methods to
        // run in accordance with which option the user selects.
        System.out.println("_____");
        System.out.println("Please input the letter of desired menu option" + "\nA: Password Checker"
                + "\nB: Password Maker" + "\nC: Quit\n");
        // Presents user with options.
        String input = scan.nextLine();
        // scans the users input storing it in a string.
        switch (input) {
        case "a":
            passIn("", false);
        case "A":
            passIn("", false);
            // calls passIn method if input is "a" or "A"
        case "b":
            passMake();
        case "B":
            passMake();
            // calls passMake method if input is "b" or "B"
        case "c":
            quit();
        case "C":
            quit();
            // calls quit method if input is "c" or "C".
        default:
            System.out.println("\nInvalid input\n");
            menu();
            // return user to menu if input is invalid.
        }
        // a switch case has been used as it is more efficient.
        // no break statement was required as each case is the switch statement calls a
        // seperate method which acts as a natural break
        // checks what the user has entered
    }

    public static void passIn(String pass, boolean gen) {
```

```
53          // This method takes the users input and determines which character types have
54          // been used in the password.
55          if (!gen) {
56              System.out.println("\nPlease input password between 8 and\n24 characters in length.\n");
57              pass = scan.nextLine();
58              // only does this if the password isn't computer generated.
59          }
60          String allowChar = "";
61          try {
62              Scanner scantxt = new Scanner(new File("allowedChars.txt"));
63              allowChar = scantxt.nextLine();
64              // stores the first line of text document as a string.
65              scantxt.close();
66          } catch (FileNotFoundException e) {
67              System.out.println(e + " Please reload program.");
68              System.exit(0);
69              // this tells the user why the program has stopped and what to do
70              // the program then terminates.
71          }
72          if (pass.length() < 8 || pass.length() > 24) {
73              System.out.println("\nPassword must be between 8 and 24 characters in length.\n");
74              menu();
75              // tells the user that the password is the wrong length and returns them to the
76              // menu.
77          }
78          HashMap<String, Boolean> charTypes = new HashMap<String, Boolean>();
79          charTypes.put("low", false);
80          charTypes.put("hig", false);
81          charTypes.put("dig", false);
82          charTypes.put("sym", false);
83          // this HashMap is for checking if each character type has been used, each key
84          // applies to a character type: low is lower case letters, hig is upper case
85          // letters, dig is digits and sym is symbols.
86          for (int i = 0; i < pass.length(); i++) {
87              boolean valid = false;
88              // stays false if the character is not allowed
89              for (int j = 0; j < allowChar.length(); j++) {
90                  if (j < 26) {
91                      if (pass.charAt(i) == allowChar.charAt(j)) {
92                          charTypes.replace("low", true);
93                          // changes the value of low to true if the password contains a lower case
94                          // letter.
95                          valid = true;
96                      }
97                  } else if (j < 52) {
98                      if (pass.charAt(i) == allowChar.charAt(j)) {
99                          charTypes.replace("hig", true);
100                         // changes the value of low to true if the password contains a upper case
101                         // letter.
102                         valid = true;
103                     }
104                 } else if (j < 62) {
105                     if (pass.charAt(i) == allowChar.charAt(j)) {
```

```
                        charTypes.replace("dig", true);
                        // changes the value of low to true if the password contains a digit.
                        valid = true;
                    }
                } else {
                    if (pass.charAt(i) == allowChar.charAt(j)) {
                        charTypes.replace("sym", true);
                        // changes the value of low to true if the password contains an allowed symbol.
                        valid = true;
                    }
                }
            }
            if (valid == false) {
                System.out.println("\nPassword contains an invalid character.\n");
                menu();
                // if the character is invalid then the user is told this and returned to the
                // menu.
            }
        }
        Boolean letters = false;
        // letters is true if only upper and lower case letters have been used.
        int numCharTypes = 0;
        if (charTypes.get("low")) {
            numCharTypes += 5;
        }
        if (charTypes.get("hig")) {
            numCharTypes += 5;
        }
        if (numCharTypes == 2) {
            letters = true;
        }
        if (charTypes.get("dig")) {
            numCharTypes += 5;
        }
        if (charTypes.get("sym")) {
            numCharTypes += 5;
        }
        if (numCharTypes != 2) {
            letters = false;
        }
        // for each character type used numCharTypes used increases by five so it can be
        // easily added to the total point score later.
        consec(gen, numCharTypes, pass, letters);
    }

    public static void consec(boolean gen, int numCharTypes, String pass, boolean letters) {
        // This method checks if the passowrd contains any sets of three characters and
        // deducts points for them.
        ArrayList<String> consecStrings = new ArrayList<String>();
        try {
            Scanner consecScan = new Scanner(new File("allowedChars.txt"));
            while (consecScan.hasNext()) {
```

```
158                 consecStrings.add(consecScan.nextLine());
159             }
160             // while there is another line in the text document the ArrayList has the value
161             // of the next line in the text document added to it.
162             consecScan.close();
163         } catch (FileNotFoundException e) {
164             System.out.println(e + "Please reload program");
165             System.exit(0);
166         }
167         // if the file doesn't exist then it throws an error, if this happens the user
168         // is told and the program shuts down.
169         int consecutives = 0;
170         for (int i = 0; i < (pass.length() - 2); i++) {
171             String passConsec = pass.charAt(i) + "" + pass.charAt(i + 1) + "" + pass.charAt(i + 2);
172             for (int j = 1; j < consecStrings.size(); j++) {
173                 if (passConsec.equalsIgnoreCase(consecStrings.get(j))) {
174                     consecutives += 5;
175                 }
176             }
177         }
178         // This adds 5 to the integer consecutives for each time a sequence of three
179         // consecutive character on a QWERTY keyboard is used in the password, so this
180         // can be easily subtracted from points total later.
181         points(gen, pass, numCharTypes, consecutives, letters);
182     }
183 
184     public static void points(boolean gen, String pass, int numCharTypes, int consecutives, boolean letters) {
185         int points = (0 - consecutives) + (numCharTypes + pass.length());
186         // the passwords point total is calculated
187         if (numCharTypes == 20) {
188             points += 10;
189             // if all 4 character types are used then the points total has 10 added to it.
190         } else if (letters == true) {
191             points -= 5;
192         } else if (numCharTypes == 5) {
193             points -= 5;
194         }
195         // if only one character type or only letters have been used then 5 points are
196         // subtracted from the passwords point total.
197         output(gen, pass, points);
198     }
199 
200     public static void output(boolean gen, String pass, int points) {
201         // This method checks the password strength and if it was computer generatesd
202         // before displaying the relevant information to the user.
203         if (gen == true && points > 20) {
204             gen = false;
205             System.out.println("\nyour password is " + pass + " it scores " + points + "\n");
206             menu();
207         }
208         // if the password is computer generated and strong then the user is shown the
209         // password and point score before being returned to the menu.
```

```
210         if (gen == true) {
211             gen = false;
212             passMake();
213         }
214         // if the password is computer generated but not strong then the password
215         // generation and checking runs again until the generated password is strong.
216         System.out.println("\npassword scores " + points);
217         if (points > 20) {
218             System.out.println("\nPassword is strong\n");
219         } else if (points < 0) {
220             System.out.println("\nPassword is weak\n");
221         } else {
222             System.out.println("\nPassword is medium\n");
223         }
224         menu();
225         // if the password was user entered then the user is shown the password score
226         // and strength then returned to the menu.
227     }
228 
229     public static void passMake() {
230         // This method generates a password if the user selects option B in the menu.
231         String pass = "";
232         String allowChar = "";
233         try {
234             Scanner scantxt = new Scanner(new File("allowedChars.txt"));
235             allowChar = scantxt.nextLine();
236             // stores the first line of text document as a string.
237             scantxt.close();
238         } catch (FileNotFoundException e) {
239             System.out.println(e + " Please reload program.");
240             System.exit(0);
241             // this tells the user why the program has stopped and what to do
242             // the program then terminates.
243         }
244         int length = (int) ((Math.random() * 5) + 8);
245         // generates random number between 8 and 12 to be used as the length of the
246         // generated password.
247         for (int i = 0; i < length; i++) {
248             pass = pass + (allowChar.charAt((int) (Math.random() * allowChar.length())));
249         }
250         // the password is the length that was generated before and each character is
251         // selcted at random from the list of allowed characters.
252         passIn(pass, true);
253         // the password is checked and the value of gen is set to true.
254     }
255 
256     public static void quit() {
257         System.out.println("\nThank you for using password checker");
258         System.exit(0);
259         // If option c is selected in the menu then the user is thanked for using the
260         // program and the program terminates.
261     }
262 }
```

Testing

Test NO	Test	What should happen	What does happen	Type of data	Pass/Fail
1	Is the menu displayed when run is pressed	The menu should be displayed	The menu is displayed	Normal	Pass
2	Does the program go to option a when “a” is entered	The program should output Checker	The program outputs Checker	Normal	Pass
3	Does the program go to option a when “b” is entered	The program should output Generator	The program outputs Generator	Normal	Pass
4	Does the program go to option a when “c” is entered	The program should output Thank you for using password checker and quit	The program output ”Thank you for using password checker” and quits	Normal	Pass
5	Does the program output invalid and return to the menu when “d” is entered	The program should output “Invalid input” and return to the menu	The program output “Invalid input” and returns to the menu	Erroneous	Pass
6	Does the program check the password “1234567890” correctly? (digits)	The program should output “dig” and “one” and “length = 10” then say the Password is medium and scores 10	The program outputs “dig” and “one” and “length = 10” then says the Password is medium and scores 10	Normal	Pass
7	Does the program check the password “aaamAM10!!” correctly, this checks if the program is distinguishing between the different character types correctly	The program should output “low” and “hig” and “dig” and “sym” and “all” and “length = 10” and the password is strong and scores 40	The program outputs “low” and “hig” and “dig” and “sym” and “all” and “length = 10” and the password is strong and scores 40	Boundary	Pass

	(Checks if having all char types is correctly checked				
8	Does the program check the password “QwErTyUiOP” correctly? (This checks if the computer recognises sequences of three consecutive characters on a QWERTY keyboard and subtracts points for them)	The program should output “low” and “hig” and “letters” and “Consecutives *8” and “length = 10” and the password is weak and scores -25	The program outputs “low” and “hig” and “letters” and “Consecutives *8” and “length = 10” and the password is weak and scores -25	Normal	Pass
9	Does the program check the password “qwertyu” correctly? (Checks for password length boundaries)	This password is 7 characters long so the program should tell the user that the password is the wrong length	The program outputs Password must be between 8 and 24 characters in length	Boundary	Pass
10	Does the program check the password “qwertyuipqwe rtyuiopqwert” correctly. (Checks for password length boundaries)	This password is 25 characters long so the program should tell the user that the password is the wrong length	The program outputs Password must be between 8 and 24 characters in length	Boundary	Pass
11	Does the program check the password “aaaaaaaa” correctly? (Checks for password length boundaries)	This password is 8 characters long so should be accepted by the program which will output “low” and “one” and “length = 8” and the password is	The program outputs “low” and “one” and “length = 8” and the password is medium and scores 8	Boundary	Pass

		medium and scores 8.			
12	Does the program check the password “aaaaaaaa aaaaaaaa aaaaaaaa” correctly? (Checks for password length boundaries)	This password is 24 characters long so should be accepted by the program which will output “low” and “one” and “length = 24” and the password is strong and scores 24	“low” and “one” and “length = 24” and the password is medium and scores 24	Boundary	Pass
13	Does the program correctly recognise invalid characters being used in the password “a a”? (Space character is used)	The program should output that the password contains an invalid character and return to the menu	The program outputs that the password contains an invalid character and returns to the menu	Erroneous	Pass
14	Does the program generate a strong password?	Example of a strong password should be outputted	The program outputs a strong password	Normal	Pass

Test 1/2

```
Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit

a
Checker
```

Test 3

```
Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit

b
Generator
```

Test 5

```
Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit

d

Invalid input


Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit
```

```
Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit

c

Thank you for using password checker
```

Test 6

```
1234567890
dig
one
length = 10

password scores 10

Password is medium


Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit
```

Test 7

```
aaamAM10!!
low
hig
dig
sym
all
length = 10

password scores 40

Password is strong


Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit
```

Test 8

```
QwErTyUiOP
low
hig
letters
Consecutives *8
length = 10

password scores -25

Password is weak


Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit
```

Test 9

```
qwertyu

Password must be between 8 and 24 characters in length.


Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit
```

Test 10

```
qwertyuiopqwertyuiopqwert

Password must be between 8 and 24 characters in length.


Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit
```

Test 11

```
aaaaaaaa
low
one
length = 8

password scores 8

Password is medium


Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit
```

Test 12

```
aaaaaaaaaaaaaaaaaaaaaaaa
low
one
length = 24

password scores 24

Password is strong


Please input the letter of desired menu option
A: Password Checker
B: Password Maker
C: Quit
```

Test 14

```
Please input the letter of desired menu optio
A: Password Checker
B: Password Maker
C: Quit

b
low
low
dig
sym
all
length = 9

your password is F%ELJcsM3 it scores 39


Please input the letter of desired menu optio
A: Password Checker
B: Password Maker
C: Quit
```

Evaluation

My program performs all the requirements of the task as demonstrated in the testing section of my report.

The code displays the user a menu, the user then selects the option they want.

If password checker is selected then the user is asked to input a password between 8 and 24 characters in length.

If the password is too long, too short or contains an invalid character then the user is told that and returned to the menu.

If the password is valid then the program correctly calculates the password score and therefore strength in accordance with the parameters given in the task before returning the user to the menu.

If the user selects Generate Password then the program outputs a password that is strong in accordance with the parameters given in the task.

If the user selects Quit then the program displays a “Goodbye” message and terminates.

To improve, I would add messages when the user is told the score of their password telling them how to improve their password for example:

If the password is only 10 characters long then the program would output “Password could be improved by increasing password length”.

If the password does not contain all character types then the user could be told that “Password could be improved by using more character types.”

If the password contains one or more sequences of 3 characters that are consecutive on a keyboard then the user could be told that “Not using a one or more sequences of 3 characters that are consecutive on a keyboard will improve the password.”

Get help and support

Visit our website for information, guidance, support and resources at aqa.org.uk/8520

You can talk directly to the Computer Science subject team

E: computerscience@aqa.org.uk

T: 0161 957 3980

aqa.org.uk

G018899