CS 210 – FALL QUARTER 2025- TAESIK KIM – TAESIK.KIM@BELLEVUECOLLEGE.EDU

LINE BREAK - 6 pt

– ASSIGNMENT 1 –

Creating User ID

(Reading/Writing a file + ArrayList)

DUE: Friday **10/31/2025**, 11:59 PM

"The world will know you as a CS210 student,
but you will be much more than that.
You will be a coding error demon hunter."

- K-pop Demon Hunters

(40 pts) - Many online and offline services are associated with a user identity and credential. In this assignment, you will create a rudimentary (and highly insecure) database(data file) for the storage of only the user generated identities. This program will have the following behaviors:

A) (3 pts) - Prior to prompting for a new user ID, the existing list of user IDs should be read and loaded to **an arrayList**.
1) Read the list from "IDs.txt", a prompt is not necessary. Do not change this data file name "IDs.txt".
2) The existing list of user IDs should be displayed to the console (screen) according to this format.
The first line displays the title and current number of IDs. From the second line, the user ID is displayed.

eg)
```
List of User IDs(3)
April#4$
Mar@33#
June6@!
```

B) (8 pts) - Upon prompting, the following checks must be made against an attempt to create a new user ID.

1) Duplicate IDs should be disallowed, and the user IDs are case sensitive.
2) IDs must be between 5 and 11 characters in length.
3) IDs must have lower-case AND upper case.
4) IDs must start with a letter like [a-z A-Z]
5) IDs must have at least one numbers.
6) IDs must have at least two special character.
7) IDs must end with a special character.

8) If the rules from 1-7 are violated, an appropriate error message with the ID should be displayed. (See example below).
i) The error message should display *all* the reasons why the ID is not correct.
ii) Do not display merely the first problem the program identifies (unless, of course, the first problem is the only problem).

9) If the rules from 1 to 7 pass,

i) You should display an appropriate message with the new ID. (see example below)
ii) "IDs.txt" datafile should be updated
iii) Read IDs.txt and display all IDs including current number of IDs on the screen. The format should be as described in the sample log below.
iv) stop the program.

C) (8 pts) – Restrictions

1) ID consists of only one token. The program doesn't need to verify that it's a single token.
2) You should create seven methods to check each requirement, and they must return true or false.
3) When checking the length of an ID, "magic number" is NOT allowed.
4) When checking the letter, number and special character of an ID, "ASCII code" is not allowed.
5) "Bad boolean zen" is NOT allowed.
6) You must use Character.isDigit(c) to check if c is a digit(=number), Character.isLetter(c) to check if c is a letter, Character.isUpperCase(c) to check if c is an uppercase and Character.isLetterOrDigit(c) to check if c is a letter or digit.

** Your code should include consistent indentation and **functions MUST be limited to those presented in chapters 1 to 7 of the textbook and the lecture notes, or you will be heavily penalized.**

D) (4 pts) – You have to create total 11 methids: a main method, one to read the file, one to write the output, 7 methods to check rules 1 through 7, and to print error messages. These 7 methods must return a boolean value.

E) (4 pts) - All error messages should be stored in an array in advance and printed using the index of that array as needed.

F) (4 pts) - When a valid ID is provided, this ID **should be written to an arrayList** and **displayed back to console(screen).**

G) (9 pts.) - Finally, the **"IDs.txt" datafile should be updated** such that the new user ID appears should the program be run again. Remember this output file is the same as the input file.

H) (-30 pts (penalty)) - Include appropriate program documentation and formatting including:

1) Your first and last name
2) Your BC email address
3) Your student ID number
4) The date
5) A short description of the program's function
6) Comments necessary to explain the operation of your program
7) Proper indentation

I) Save your file as "UserIDs.java" and **upload** to Canvas. (MUST **UPLOAD** UserIDs.java file. NOT copy/pase images)
You don't need to upload IDs.txt.

Sample log of execution is as follows (user inputs are in **bold and underlined**):

(line break, 11 pt)

```
List of User IDs(3)
April#4$
Mar@33#
June6@!
```

Create a new ID: **Mar@33#**
Mar@33# is alread in use.

Create a new ID: **December!!!!**
December!!!! must be between 5 and 11.
December!!!! must have at least one number.

Create a new ID: **bellevuecollege**
bellevuecollege must be between 5 and 11.
bellevuecollege must have at least one number.
bellevuecollege must have at least two special characters.
bellevuecollege must have lower-case and upper-case.
bellevuecollege must end with a special character.

.
.

Create a new ID: **July04!!**
July04!! created successfully!

List of IDs(4)
April#4$
Mar@33#
June6@!
July04!!