CIS611
Individual Practice Programming Assignment: PA04
Total Points: 20

Methods, Sorting, and I/O Files:

The purpose of this programming assignment is to:

- I/O File operations
- Use predefined file classes to read and write data to files
- Use sorting algorithms (methods) to sort data
- To develop reusable code for modularity, readability and maintainability
- Define methods and invoke them
- Use method overloading

Q1 (10 points)**:**

Some Websites impose certain rules for passwords. Write a Java program that has a main method and another method to check whether a string is a valid password. Suppose the password rule is as follows:

- A password must have at least eight characters.
- A password consists of any sequence of letters and digits, as well as at least one special characters. You will code (and test) for only these three special characters, which are %, &, #. You are not required to code to account for any other special character, other than these three.
- A password must contain at least two digits.

Write a program that has a code in the main method to prompt the user to enter a password and calls another static method (`public static boolean isValidPassword(String s)`) to check the password and displays "valid password", if the rule is followed; or "invalid password" otherwise. The `isValidPassword()` method applies the password rules; it returns true if the password format is correct, otherwise it returns false. You should use the JOPtionPane class to interact with the user. The program continues to accept user inputs (passwords) as long as the user responds with Yes to the JOPtionPane Confirm Dialog message.

Sample 1
Enter a string for password: wew%ew43
valid password

Sample 2
Enter a string for password: 343a\
invalid password

Hint: you may use some of the predefined **Character** class methods, such as `Character.isLetter()` and `Character.isDigit`

Q2 (10 points)**:**
Create a Java project that has two classes, the main entry **Product** and **Sort** classes. The program reads data from a text file, sort the data using the selection sort algorithm, and then store the sorted data in a different text file. The data in the text file is sorted based on the product names, then it should be sorted based on the product prices, and finally sorted data is stored in a text file.
The Product class has the following methods:

- The *main()* method, which prompt the user to input the file name of the input file (this may include the file path if the file is not stored in the same project folder), creates two arrays pName (String[]) and pPrice (double[]) of size 50, and then sequentially calls and passes file name (path) pName, pPrice to the static methods, *readFromFile(), sortArrays*, and *writeToFile()*
- *readFromFile()* is a static method that will read data from the enclosed "products.txt" text file with this document, and it stores the product names and product prices in the method parameters arrays pName and pPrice, respectively. After complete reading data from file, the method should display a JOPtionPane message dialog of the array elements (product names and prices)
- *sortArrays()* is a static method that passes the parameter arrays pName and pPrice to the static *selectionSort()* method in the **Sort** class in order to sort both arrays based on the prices data elements in the pPrice array, that means any change in the pPrice array will also results in a change in the pName array.
- *writeToFile()* is a static method that will write/store the sorted arrays data elements in the parameter list (pName and pPrice) into a file (line by line), the data should be stored in the "sortedProducts.txt" text file. After complete writing data to file, the method should display a JOPtionPane message dialog of the array elements (product names and prices) , data should be sorted based on the product prices

The **Sort** class has only one static *selectionSort()* method that receives two arrays in its parameter list (pName and pPrice) and sorts the arrays by using the selection sort algorithm. It basically sorts the pPrice array in an ascending order, so that any change in pPrice array results in a change in the pName array in order to keep the product name and price elements in both arrays in the same order (having the same index values for each name and price in both pName and pPrice arrays)

Evaluation Criteria:

- The programs must compile cleanly (no compile errors, but compile warnings are sometimes accepted)
- The program should handles invalid data inputs by users and terminates gracefully
- The programs should not crash while running and it should terminate
- All tasks (requirements) in this assignment must be completed in order to receive credit
- The correct understanding and implementation (coding) of the requirements (programs should behave as anticipated):
    - The programs must terminate with proper/correct outputs
    - All the logical computations should be performed correctly

Submission: (This is an individual Assignment!)

Copy the .java source files from the *src* folder in your *work space* to another folder that should be named following the provided naming format in this course, then zip and upload the file under this assignment answer in Canvas.

File Name: *FLLLLPA04.zip (F = first letter in your first name and LLLL = your last name)*

Grading Rubric - PA04

Student Name: _____

Question 1

Requirements	Any comment provided by grader	Max Points Allowed	Points Earned
General Code Structure: Proper naming convention used for file (0.25) Comments used in the code to explain the purpose of the code (0.25) Indentation of the code for better readability (0.25) Good choice of variable names (0.25)		1	
Input, Output, User Interface: Proper coding implementation of the logic to read the data (1) Proper coding implementation of displaying the expected output (1) Exception handling of the invalid input values. For example if no value is entered, or empty space is entered, or invalid data is entered, the program should not crash (2) Continue to accept use input when the user presses Yes (1)		5	

General Algorithm and Logic: Proper implementations of password rule and the method `isValidPassword` (3) Use of Character class (1)		4	
Total		10	

Question 2

Requirements	Any comment provided by grader	Max Points Allowed	Points Earned
General Code Structure: Proper naming convention used for file (0.25) Comments used in the code to explain the purpose of the code (0.25) Indentation of the code for better readability (0.25) Good choice of variable names (0.25)		1	
Input, Output, User Interface: Proper coding implementation of the logic to read the data from the text file (1.5) Proper coding implementation of writing the expected data to another text file (1.5) Exception handling of the invalid input values. For example, if no value is entered, or empty space is entered, invalid data is entered, the program should not crash (1)		4	
General Algorithm and Logic: Use of array of data type String to hold name (0.5) Use of array of data type double to hold price (0.5) Proper coding implementation of the methods readFromFile, sortArrays, selectionSort, writeToFile (4)		5	

Total		10	

Total ____/20