LAB211 Assignment

Type:	**Long Assignment**
Code:	**J1.L.P0032**
LOC:	**500**
Slot(s):	**N/A**

Title

Car Showroom Management

Background

A car showroom named *Michael BMW* maintains a list of BMW cars. The showroom data is stored in two text files:

- **brands.txt** – containing BMW brand information
- **cars.txt** – containing car information

File brands.txt	Description
B7-2018, BMW 730Li (2018), Harman Kardon: 3.749B B7-MS, BMW 730Li M Sport, Harman Kardon: 4.319B B7-MS20, BMW 730Li M Sport (2020), Harman Kardon: 4.369B B7-PE, BMW 730Li Pure Excellence, Harman Kardon: 4.929B B5-18, BMW 530i (2018), Alpine: 2.599B B7019, BMW 530i (2019), Alpine: 2.729B BX4-18, BMW X4 xDrive20i (2018), Sony: 2.799B BX4-17, BMW X4 xDrive20i (2019), Sony: 2.899B B3-GT18, BMW 320i GT (2018), Bose: 1.799B B3-S19, BMW 320i Sportline (2019), Bose: 1.899B B5-X19, BMW X5 xDrive40i XLine (2019), Bose: 4.199B B5-X20, BMW X5 xDrive40i XLine (2020), Bose: 4.239B	Information on a line: <Brand ID, brand name, sound brand: price> B: Billion

File cars.txt	Description
C01, B7-2018, red, F12345, E12345 C02, B7-2018, black, F12346, E12346 C03, B7-MS, orange, F12347, E12347 C04, B7-MS20, white, F12348, E12348 C05, B7-PE, pink, F12349, E12349 C06, B5-18, pink, F12350, E12350 C07, B5-X20, grey, F12351, E12351	Information on a line: <Car ID, brand ID, color, frame ID, engine ID>

Constraints

1- **Constraints on brands**:
   a. Brand ID must be unique.
   b. Brand name cannot be empty.
   c. Sound manufacturers cannot be empty.
   d. Price must be a positive real number. For example, 1.234 means that 1.234 billion(s)$

2- **Constraints on cars**:
   a. Car ID cannot be unique.
   b. Brand ID must exist in the brand list.
   c. Color cannot be empty.
   d. Frame ID cannot be empty, must follow the format “F00000”, and must be unique.
   e. Engine ID cannot be empty, must follow the format “E00000”, and must be unique.

Program Specifications

Students are required to build a **Java console application** using the **OOP approach** to manage brands and cars in the showroom.

The program must implement the following functions:

1- List all brands
2- Add a new brand
3- Search for a brand by ID
4- Update a brand by ID
5- List all brands with prices less than or equal to an input value
6- List all cars in ascending order of brand names
7- Search cars by partial brand name match
8- Add a new car
9- Remove a car by ID
10- Update a car by ID
11- List all cars by a specific color
12- Save data to files
13- Quit program

Features and Assessment:

No.	LOCs	Function
1/	20	**List all brands**: Display data in a table with column headers.
2/	50	**Add a new brand**: Accept brand ID, name, sound system brand, and price; add to the list if constraints are satisfied. Display success/failure message.
3/	20	**Search for a brand by ID**: If the brand does not exist, show: *"This brand does not exist!"*. Otherwise, display brand information.
4/	50	**Update a brand by ID**: If the brand does not exist, show: *"This brand does not exist!"*. Otherwise, allow updating brand name, sound system brand, and price. Empty input means skipping the update for that field.
5/	50	**List all brands with prices less than or equal to an input value**: Accept a price input; display all brands with price ≤ input.
6/	20	**List all cars in ascending order of brand names**: Display all cars in ascending order of brand names; if same brand, sort by price descending.
7/	80	**Search cars by partial brand name match**: Accept partial brand name (e.g., *320i*); display all matching cars.
8/	80	**Add a new car**: Accept car ID, brand ID (via menu), color, frame ID, and engine ID. Validate constraints.
9/	20	**Remove a car by ID**: Accept car ID. If the car does not exist, show: *"This car does not exist!"*. Otherwise, remove it.
10/	50	**Update a car by ID**: Accept car ID. If the car does not exist, show: *"This car does not exist!"*. Otherwise, allow updating color, frame ID, and engine ID. Empty input means skipping the update for that field.
11/	20	**List all cars by a specific color**: Accept a color and display all cars of that color.
12/	20	**Save data to files**: Save both brand and car lists into files. Show confirmation message after completion.
13/	20	**Quit program**: If changes were made to brands or cars, save updated lists before exiting.

Note for Students:

- ***Follow OOP principles*** *(encapsulation, inheritance, polymorphism, abstraction).*

- ***Follow Computational*** *Thinking (Decomposition, Pattern Recognition, Abstraction, and Algorithm) for reporting and presentation.*
- ***Input validation must strictly follow constraints.***
- ***Show appropriate messages for invalid or missing data.***
- ***Evaluation: Understanding, Correctness, data validation, and coding style.***
- ***Submission***
    - ***Source code***
    - ***Diagram***
    - ***Flow-chart***