

# CSC10012

## LAB 06 – FILE HANDLING

FIT-HCMUS

### Part 1 – Text File

#### Problem 1

Assume a car is represented by the following information:

- Make: A string representing the car manufacturer, without any spaces (e.g., Ford, Honda)
- Model: A no-space string representing the car model, without any spaces (e.g., Taurus, Accord)
- Year: An integer representing the year of manufacture
- Cost: A floating-point number representing the cost of the car
- Quantity: An integer representing the number of cars in stock

The file "data-1-1.txt" contains sample car data. Below are some lines from the beginning of the file:

```
Make Model Year Cost Quantity
Ford Taurus 2023 32000 5
Honda Accord 2024 28500 3
BMW 3-Series 2023 42000 1
Mercedes-Benz E-Class 2022 55000 0
Tesla Model-3 2024 40000 10
```

, in which:

- The first line lists the information fields included in the data.
- For the next lines, each subsequent line represents the information for a car, separated by spaces.

Given the `Car` struct as follows:

---

```
struct Car {
    char carMake[16];
    char carModel[16];
    int yearModel, quantity;
    double cost;
};
```

---

Write a program to manage cars, including the following functionalities.

1. Read Database: Read the list of all cars from the file "data-1-1.txt".
2. Display Inventory: Print a list of all cars.
3. Add a new car.
4. Sell Car: If the requested car is available in sufficient quantity, sell it and update the inventory. Otherwise, inform the user that either the car is not available or there's an insufficient quantity.
5. Remove Car: Remove a car from the inventory.
6. Track Sales: Record all sold cars. Create a function to calculate total sales for a specific month.
7. Update Database: Overwrite "data-1-1.txt" with the latest changes the program made.

## Problem 2

Assume a player is represented by the following information:

- Name: A string representing the player's name
- Number of assists: An integer representing the number of assists the player has made
- Number of goals: An integer representing the number of goals the player has scored

The file "data-1-2.txt" contains sample player data. Below are some lines from the beginning of the file:

```
Name,Number of assists,Number of goals
Erling Haaland,10,35
Kevin De Bruyne,25,15
Jack Grealish,12,10
Phil Foden,15,20
Bernardo Silva,18,8
```

, in which:

- The first line lists the information fields included in the data.
- For the next lines, each subsequent line represents the information for a player, separated by commas.

Given the `Player` struct as follows:

---

```
struct Player {
    char name[16];
    int numAssists, numGoals;
};
```

---

Write a program to manage players in a football club, including the following functionalities:

1. Read Database: Read list of all players in the club from the file "data-1-2.txt".
2. Display Players: Print a list of all players currently in the club.
3. Add Player: Add a new player to the club.

4. Remove Player: Remove a player from the club.
5. Update Player: Update a player's statistics (goals and assists).
6. Find Best Player: Determine the best player for the season based on a calculated score. The best player score is calculated using the formula:  $\text{best player score} = 0.4 \times \text{number of assists} + 0.6 \times \text{number of goals}$ .
7. Update Database: Overwrite "data-1-2.txt" with the latest changes the program made.

## Part 2 – Binary File

Given the binary file "data-2-1.bin" and "data-2-2.bin" are created by dumping the full list of cars and players from the texts file "data-1-1.txt" and "data-1-2.txt". Maintain the same functionalities as described above for each problem. However, for problem 1, instead of reading data from the text file "data-1-1.txt", students will read information from the binary file "data-2-1.bin". Similarly, for problem 2, read data from the binary file "data-2-2.bin".