Lab 4

PART A – In-Class (Due End of Class), 3 marks

1. Create a Python Project with the name Lab4
2. Create a Python file called lab4.py
3. In file lab4.py Create function main in the format:

```
def main():
    pass


if __name__ = "__main__":
    main()
```

4. Create a Python file with the name login.py

   In file login.py create the following function:

- Function generate_login(), the function takes three parameters (first name, last name and BCIT ID), and returns the default login ID.
    - First and last name should be properly formatted i.e. the first character is upper case and the rest of the name is lowercase.

  The login ID is generated as follows:
    - Get the first three letters from a properly formatted first name, if the first name length is less than three characters then the entire name will be used.
    - Get the first three characters from a properly formatted last name, if the last name length is less than three characters then the entire last name will be used.
    - Get the last three characters of the BCIT ID, if BCIT ID length is less than three characters then the entire ID will be used.
    - Concatenate the characters generated from the above instructions as follows:

      (Characters from the first name + characters from the last name+ characters from BCIT ID)

      Return this concatenated string as the login id.

- In lab4.py:
    - Import module login (corresponding to login.py) and sys (built-in to Python)
    - In the main() function get the first name, last name and BCIT ID from command line arguments:
      python main.py "First Name" "Last Name" A01010101
    - Create the default login ID by passing those values to function generate_login() and display the result (using a f-string or formatted string expression). It should look like:

      Your login id is FirLas101

Demo to your instructor the output from your code above using command line arguments. Make sure to include the Part A code in lab4.py and login.py file with your Part B submission as well.

Part B – Take Home (due before next Lecture), 8 marks

This continues with the lab4.py script and login.py module.

In login.py create the following function:

- Function get_password(), this function takes in no parameters and returns a string password
    - The function prompts the user to enter a password
    - The password must meet the following criteria:
        - At least 7 characters long
        - At most 30 characters long
        - Cannot contain the text “password”
        - Must be all alphanumeric characters
        - Cannot have spaces
    - If the password does not meet the above criteria, the user should be told that their password is invalid and the reason. They should be prompted to enter the password again (hint: use a while loop)

- More than one error message on the password can be displayed at the same time (hint: distinct if statements)

Call function get_password() from lab4a.py after the default login ID is generated to prompt the user to set the initial password.

- Display the last 4 characters of password returned by function get_password. It should look like this if the password entered was *Test12345*:

  Your initial password ends with 2345

Sample output for running lab4.py after Parts A and B:

```
Your login id is Johnny001
Enter you password:short
password must be at least 7 characters
Enter you password:verylongpasswordmorethan30characte
password must be 30 characters or less
password cannot include the text "password"
Enter you password:password
password cannot include the text "password"
Enter you password:password with spaces
password cannot include the text "password"
password must be alphanumeric
password cannot have spaces
Enter you password:passwordwith$$$
password cannot include the text "password"
password must be alphanumeric
Enter you password:GoodPassword
Password is valid
Your initial password ends with word
```

Include login.py and lab4.py for your Parts A and B submission.

Part C – Take Home (due Feb. 7th at midnight), 8 marks

Create a file with the name data_format.py. This file with have a main function that calls the functions below.

Add the following functions to file data_format:

- function get_car_csv_info(), the function has no parameters but asks the user to enter the following information:
    - car make (i.e., Ford, Honda, Toyota)
    - car model (i.e., Mustang, Civic, Camry)
    - car year (i.e., 2021)
    - car mileage (i.e., 20000)
    - car price (i.e., 25444.34)

  The function will eliminate leading and trailing spaces from make, model, year, mileage and price (use string method strip ()).
  The function will set the make and model to title case (using the string method title())
  The function will return a CSV formatted string from the given information in the same order specified above, separated by commas (,). See sample output below. **You must use a string formatting expression or f-string for this.**
  Note: price should be formatted to have two digits after the decimal point
- Function to_json_format(), the function takes a CSV formatted string with the car info and returns the corresponding JSON format string.
    - Use the split string method to extra the car data from the CSV formatted string
    - Use string concatenation to generate a string with the JSON car info data. See sample output below. **Do not use a formatted string expression or f-string.**
- Create a main function as shown in the in-class lab section within the data_format.py file. It should:

- Call the get_car_info function and store the returned value in a variable
- Print the text "The car CSV data is: "and then CSV car info returned from the function (see the format in the sample below)
- Call the to_json_format function passing in the CSV car info and store the returned value in a variable
- Print the text "The car JSON data is: "and then the JSON car info returned from the function (see the format in the sample below)

All your print statements should be in this main function, there should be no print statements in the other functions.

Example of the expected input and output:

```
enter car make:   honda
enter car model: civic
enter car year:   2000
enter car mileage:   58123
enter car price:  19999.123
The CSV car data is: Honda,Civic,2000,58123,19999.12
The JSON car data is: {"make": "Honda", "model": "Civic", "year": "2000", "mileage": "58123", "price": "19999.12"}
```

Best Practices

- Variable names are descriptive and should be lower_snake_case
- function names should be lower_snake_case
- All functions include Doc-String comments

Part B Submission

Submit a zipfile with your **login.py**, **lab4.py** and **data_format.py** files to the Lab4 dropbox on the Learning Hub before next lecture.