

Final Lab Exam
Advanced Computer Programming CS 121

Instructions:

This is a major exam; the same rules apply.

Use VS Code and login to your GitHub via CLI/command prompt.

First, check the current credentials by typing in the command prompt:

```
git config --global user.email  
git config --global user.name
```

After checking, set your own email address and username by typing:

```
git config --global user.email y*****@gmail.com  
git config --global user.name Y*****
```

```
git init  
git add README.md  
git commit -m "commit message"
```

Create a repository named final_lab_exam_cs121_LASTNAME1_LASTNAME2

Exam:

A TEMPLATE CODE IS PROVIDED. THE STUDENT'S MAY OR MAY NOT FOLLOW EXACTLY THE TEMPLATE BUT MUST ADHERE TO THE REQUIREMENTS BEING ASKED. THEY MAY USE OTHER WAYS TO WRITE THE CODE BUT THE LOGIC AND FLOW SHOULD BE EXACTLY THE SAME AS PRESENTED IN THE RUNNING APP DURING THIS EXAM.

Create a dice game.

The game will provide a registration, login, and exit for its main menu. The app will show the main menu the moment it runs.

```
Welcome to Dice Roll Game!  
1. Register  
2. Login  
3. Exit
```

The menu will only accept based on the numbers in the menu; wrong inputs will tell the user "Invalid choice. Please try again."

```
Invalid choice. Please try again.
```

Registration will require a username and password. The username must be at least 4 characters while the password is 8 characters.

```
Enter username (at least 4 characters), or leave blank to cancel: e  
Username must be at least 4 characters long.
```

```
Enter password (at least 8 characters), or leave blank to cancel: r  
Password must be at least 8 characters long.
```

If the user does not want to continue or want to cancel, they can simply leave the prompt blank to return to the previous menu.

The registration should not allow usernames that already exist.

```
Enter username (at least 4 characters), or leave blank to cancel: qwer  
Enter password (at least 8 characters), or leave blank to cancel: qwerqwer  
Username already exists.
```

If the user successfully registered, something like this should be shown:

```
Enter your choice, or leave blank to cancel: 1  
Registration  
Enter username (at least 4 characters), or leave blank to cancel: zxcv  
Enter password (at least 8 characters), or leave blank to cancel: zxcvzxcv  
Registration successful.
```

Once registered, the user can now login:

```
Welcome to Dice Roll Game!
1. Register
2. Login
3. Exit
Enter your choice, or leave blank to cancel: 2
Login
Enter username, or leave blank to cancel: qwer
Enter password, or leave blank to cancel: qwerqwer
```

When logged in, a new menu should be shown, which also shows the username of the logged in user.

```
Welcome, qwer!
Menu:
1. Start game
2. Show top scores
3. Log out
```

When starting a game, the dice roll is done automatically. Each game will have a stage, where each stage will be a best of three, where if the player wins, the user can choose to continue to the next stage or stop playing the game. A tie will not count a score for anyone, allowing for an additional roll, until the best out of three condition is met. If the user loses, the "Game over. You didn't win any stages." Will be shown. It is worth noting that each game is unique.

```
Enter your choice, or leave blank to cancel: 1
Starting game as qwer...
qwer rolled: 2
CPU rolled: 2
It's a tie!
qwer rolled: 2
CPU rolled: 3
CPU wins this round!
qwer rolled: 4
CPU rolled: 4
It's a tie!
You lost this stage qwer.
Game over. You didn't win any stages.
```

In case the player wins a stage, they can choose to proceed to another stage for another game of best out of three. The user will be shown the total points earned and the number of stages won in that current game.

```
Enter your choice, or leave blank to cancel: 1
Starting game as qwer...
qwer rolled: 6
CPU rolled: 1
You win this round! qwer
qwer rolled: 6
CPU rolled: 5
You win this round! qwer
qwer rolled: 3
CPU rolled: 1
You win this round! qwer
You won this stage qwer!
qwer
Total Points: 6, Stages Won: 1
Do you want to continue to the next stage? (1 for Yes, 0 for No):
```

The user can choose 1 to continue and 0 to stop if they win. Any other entries should be handled.

```
Do you want to continue to the next stage? (1 for Yes, 0 for No): sd
Invalid input. Please enter 1 for Yes or 0 for No.
```

For the scoring, every round the player wins, they receive 1 point, and if they win the stage, they receive an additional 3 points. As the user progresses, the points during that game will stack up, until the user loses.

Remember, in every game, the starting points will always be 0.

For the top score selection, the user should be able to see the top-10 highest scores attained.

If no scores are made, they will not be able to see anything yet.

```
Welcome, qwer!
Menu:
1. Start game
2. Show top scores
3. Log out

Enter your choice, or leave blank to cancel: 2
No games played yet. Play a game to see top scores.
```

After playing at least one game and earning a win on at least 1 stage, the user will gain a record. Else, no new records will be added.

Enter your choice, or leave blank to cancel: 1

Starting game as qwer...

qwer rolled: 5

CPU rolled: 2

You win this round! qwer

qwer rolled: 2

CPU rolled: 3

CPU wins this round!

qwer rolled: 3

CPU rolled: 6

CPU wins this round!

You lost this stage qwer.

Game over. You didn't win any stages.

Menu:

1. Start game
2. Show top scores
3. Log out

Enter your choice, or leave blank to cancel: 2

No games played yet. Play a game to see top scores.

Menu:

1. Start game
2. Show top scores
3. Log out

If the user does win at least one stage, may they stop or continue, they will have that game recorded.

```
You won this stage qwer!  
qwer  
Total Points: 6, Stages Won: 1  
Do you want to continue to the next stage? (1 for Yes, 0 for No): 0  
Game over. You won 1 stage(s) with a total of 6 points.  
Menu:  
1. Start game  
2. Show top scores  
3. Log out  
  
Enter your choice, or leave blank to cancel: 2  
Top Scores:  
1. qwer: Points - 6, Wins - 1  
Menu:  
1. Start game  
2. Show top scores  
3. Log out
```

The games will record a total of top-10 best records. Anything below the current scores, regardless of the user winning, they will not get a new record.

```
Enter your choice, or leave blank to cancel: 2  
Top Scores:  
1. qwer: Points - 18, Wins - 4  
2. qwer: Points - 8, Wins - 2  
3. qwer: Points - 6, Wins - 1  
4. qwer: Points - 6, Wins - 1  
5. qwer: Points - 6, Wins - 1  
6. qwer: Points - 6, Wins - 1  
7. qwer: Points - 6, Wins - 1  
8. qwer: Points - 6, Wins - 1  
9. qwer: Points - 6, Wins - 1  
10. qwer: Points - 2, Wins - 1
```

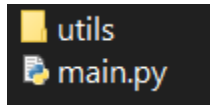
Regardless of the account logged in, they should always see the same top-10 ranking as the other users. This time, zxcv user is the one logged in, but still able to view the same top-10 ranking as user qwer.

```
Welcome, zxcv!
Menu:
1. Start game
2. Show top scores
3. Log out

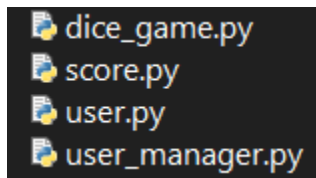
Enter your choice, or leave blank to cancel: 2
Top Scores:
1. qwer: Points - 18, Wins - 4
2. qwer: Points - 8, Wins - 2
3. qwer: Points - 6, Wins - 1
4. qwer: Points - 6, Wins - 1
5. qwer: Points - 6, Wins - 1
6. qwer: Points - 6, Wins - 1
7. qwer: Points - 6, Wins - 1
8. qwer: Points - 6, Wins - 1
9. qwer: Points - 6, Wins - 1
10. qwer: Points - 2, Wins - 1
```

Code design:

The structure should be like, the working directory will only contain the file main.py and the folder utils.



The utils folder should include the dice_game.py, score.py, user.py, and user_manager.py.



The dice_game.py holds the DiceGame class that has attributes of the user_manager object from the UserManager() class, scores placed in an empty directory, current_user set to None, and the load_score() invocation.

The load_score allows the creation of the data folder, as the game will not have it unless the driver code is run. It will create the rankings.txt file inside the data folder in the working directory.

The save_score method saves the scores based on the logic of the game into the rankings.txt where it records the username, game_id, points, and wins.

The play_game method executes the game discussed earlier.

The show_top_scores execute the scoring or ranking system discussed earlier. Make sure that the scores are sorted properly based on their points.

The logout method simply logs out the currently logged in user.

```
Welcome, qwer!  
Menu:  
1. Start game  
2. Show top scores  
3. Log out  
  
Enter your choice, or leave blank to cancel: 3  
Goodbye qwer!  
You Logged out successfully.
```

The menu method generates the menu when running the app or without a user logged in, presented in the earlier parts of this document.

In the Score class, it should hold attributes like username, game_id, points = 0, and wins = 0.

The User class should have attributes including username, password, points = 0, and stages_won = 0. This should be implemented in a Composition approach.

The UserManager can have attributes like users as an empty dictionary, and the invocation of a load_users() method.

The load_users method in the UserManager class creates a data folder if not present yet, and generates a users.txt file in it. It simply stores the username and password separated by a comma like in this example:

```
qwer,qwerqwer  
zxcv,zxcvzxcv
```

The save_users method in the UserManager class just simply saves those registered users into the users.txt file.

A validate_username and validate_password should be created based on the mentioned requirements earlier in this document.

The register and login go the same way. Utilize the User class to generate the username and password for the user that successfully passes the requirements set by the username and password validation methods.

Date time module BONUS!:

Import datetime

Strftime

The ``strftime`` function in Python stands for “string format time” and is part of the ``datetime`` module. It converts a ``datetime`` object into a string based on a specified format representing the date and time.

The `datetime` module's `now()` function returns the current time as well as the date. The `datetime.now()` function returns the current local time and date. It displays datetime in the `YYYY-mm-dd hh:mm:ss`.

```
datetime.datetime.now().strftime("%Y%m%d%H%M%S%f")
```