PRG 600

Midterm - February 15, 2023 08:00 AM - 9:45 AM

Worth 20% of final grade [20 marks + 5 Bonus Marks]

Submission Instructions

- Update the doc string at the top of the source code file called midtermprg600.py.
- Files must be submitted as individual files NOT Zipped.
- Any answers to the question and screenshots of manual test run to produce required output along with any explanations must be provided in the pdf file named as “answers_username.pdf”. Sample answers_username.docx file can be downloaded from Blackboard. Ensure the final pdf has all the screenshots in appropriate sections and are clearly visible within the page.

Midterm Package

1. midtermprg600.py
2. sample_output.txt
3. testmidterm.py
4. answers_username.docx
5. midtermprg600-check.txt

Required in Submission

If you using the functions breakdown provided

1. midtermprg600.py (Completed file)
2. Midterm600-check.txt (Check script generated file)
3. answers_username.pdf (Answers Document (PDF) change username to yourname)
4. output_username.txt (Terminal output of program run with same values as sample_output.txt, change username to yourname)
5. Screen recording of testing the code manually (Optional)

If you using your own design and organisation of functions and code

1. midtermprg600.py (Completed file - your own design and organisation of functions and code)
2. answers_username.pdf (Answers Document (PDF) change username to yourname)
3. output_username.txt (Terminal output of program run with same values as sample_output.txt, change username to yourname)
4. Screen recording of testing the code manually

How do I attempt the test

1. Read the Test Manual
2. Answer question-one on flowchart and add your image of flowchart in one page in answers_username.pdf
3. Start coding and complete the functions in midtermprg600.py
4. Once completed run manual tests as required using the sample_output.txt as guidance
5. Copy your manual test run output from the terminal into output_username.txt. (DO NOT REMOVE the ERRORS - IF YOU SEEING ANY - THIS IS IMPORTANT AS YOU COULD GET PARTIAL POINTS IN SOME CASES)
6. Run the testmidterm.py script to validate your code
7. If your code is not complete or has errors you might see some tests failed, in which case take necessary screenshots from your manual testing and the testmidterm.py run output and provide them under appropriate question number.
8. You should also consider doing screen recording of your manual testing and upload that video in addition to the screenshots.
9. Your should now have all required files to submit as mentioned at the beginning of this document
10. Rename the “username” in the file name into your own username and submit them (WATCH OUT FOR FILE EXTENSION)
11. If you are not following the functions breakdown and code design guide provided. You can choose to factor your code however you want. Ultimately you should achieve the same result but without using any external libraries. In which case you should provide a screen recording (Mandatory) for manual testing with full coverage creating the expected output as per the guidance in sample_output.txt and any other cases you have implemented. You will also need to submit other required files.

Interview:

The professor can call any student for a one on one interview to explain the code and answer some questions about the code. Failing to participate in this interview will result in a grade of zero in the assignment.

General Description:

In this test you will be writing a python code which simulates a simple dice rolling game. It will be a multi-round game which works as the following.

Game Setup

Before the game starts, the user is asked to choose the number of players that are going to play the game and respectively asks the names of each player. So, if the user selects 4 players, the game will ask for 4 players’ names and save them, respectively.

After successfully setting up the players, the user is prompted to enter the number of rounds the players wish to play. Once this is completed, the game starts with an empty scoreboard. An example of a scoreboard is shown below:

```
****************** Round 1 ******************
        Round 1   Round 2   Round 3   Total
Appla     0         0         0         0
Applb     0         0         0         0
Applc     0         0         0         0
Appld     0         0         0         0
*********************************************
```

Playing the game

After the game has saved the names of the players and knows how many rounds it will be played for, the game starts. During each round, each player (virtually) rolls two dice and the sum of the two dice is saved for each player per round. Players will roll the dice in the same order that their names were entered during the set up. Rolling the dice is as easy as the following.
The program prints the name of the player that must play and asks them to hit enter to roll their dice and the program generates two random numbers between 1 and 6 for each dice.

```
Appla! Hit enter once you are ready to roll your dices!
2 and 6
Applb! Hit enter once you are ready to roll your dices!
5 and 6
Applc! Hit enter once you are ready to roll your dices!
3 and 6
Appld! Hit enter once you are ready to roll your dices!
5 and 3
```

After each round

At the end of each round, the game shows the scoreboard containing each player's scores per round and their respective total scores.

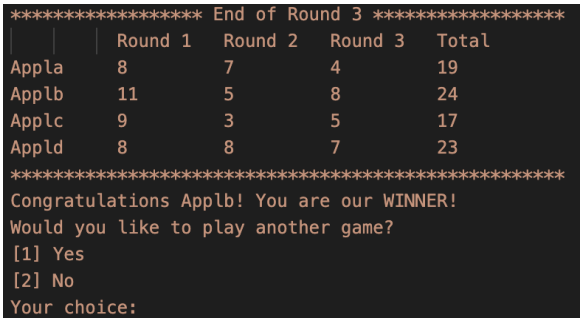
```
****************** End of Round 1 ******************
        Round 1   Round 2   Round 3   Total
Appla     8         0         0         8
Applb     11        0         0         11
Applc     9         0         0         9
Appld     8         0         0         8
*****************************************************
```

```
****************** End of Round 2 ******************
        Round 1   Round 2   Round 3   Total
Appla     8         7         0         15
Applb     11        5         0         16
Applc     9         3         0         12
Appld     8         8         0         16
*****************************************************
```

After the last round

Once the last player has played the last round, the game shows the final scoreboard, congratulates the winner, and asks whether the players want to play another game. If the players choose to play again, the game starts over. If more than one are winning (that is

equal points) all the users will be shown comma separated (Eg: Congratulation John, Kate! You are our WINNER!)



Specific Instructions

- You CANNOT use third-party libraries in your code. Python3 libraries are sufficiently enough to accomplish this assignment.
- A comprehensive output along with docstring for the functions are provided in the midterm.py code.
- This gives you a preliminary understanding of the functions, expected behaviour and expected output.
- sample_output.txt file has a full comprehensive output expected when the final product is executed locally.

Hint

- In some cases you will have to write function 1 correctly in order for function 2 to work. Making one mistake in one function possibly makes function 2 not work or provide errors. In such cases consider the game as a 2D list with each element in the game as rows (players) and the nested lists as columns (rounds).

  For example:
  game = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]

  [1,2,3,4] considered scores for 4 rounds for player 1
  [5,6,7,8] considered scores for 4 rounds for player 2
  [9,10,11,12] considered scores for 4 rounds for player 3

  This can be visualised as a table shown below:

	Round 1	Round 2	Round 3	Round 4
Player 1	1	2	3	4
Player 2	5	6	7	8
Player 3	9	10	11	12

If you run into issues you can initiate the game list as above manually and continue to develop your subsequent functions. As this way in case one function is causing another function to error you can still get partial credit.

Generating output_username.txt

The midterm package has sample_output.txt file the file contains successful execution of the program.
The first 18 lines are below:

```
Please enter the number of players: Four
Invalid input please try again
Please enter the number of players: 4
Please enter player #1 name: Appla
Please enter player #2 name: Applb
Please enter player #3 name: Applc
Please enter player #4 name: Appld
Please enter how many rounds the players wish to play: Three
Invalid input please try again
Please enter how many rounds the players wish to play: 3
****************** Round 1 ******************
        Round 1   Round 2   Round 3   Total
Appla     0         0         0         0
Applb     0         0         0         0
Applc     0         0         0         0
Appld     0         0         0         0
*********************************************
```

Once your application is complete run your program (Eg: python3 midtermprg600.py) and provide the same values as above to generate your output.

Note: Since the asktoroll will be producing random numbers your outputs will be different after line 19.

After running the game 2 times and quitting, copy the screen output into a file called “output_username.txt” and submit it as part of your submission.

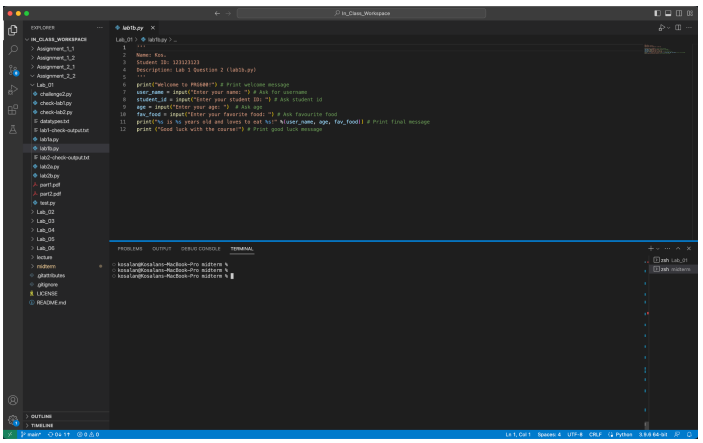
What if my Check Script Failing

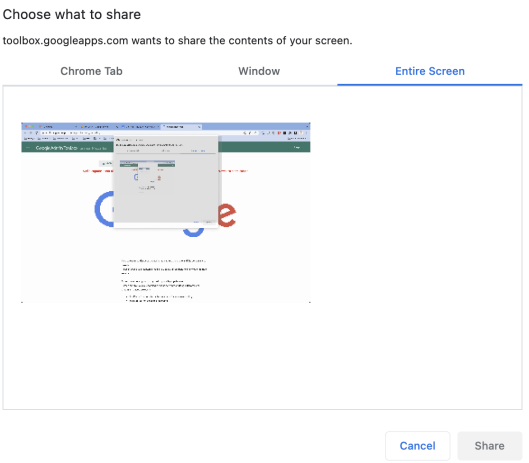
If the check scripts are failing in certain functions you only need to provide screenshots for that specific function, however if all the functions are failing provide screenshots for all.

- Screenshot of error you see in terminal when running the checkscript
- Successful test coverage screenshots, covering the sample input and output from the sample_output.txt. Refer to how to generate output_username.txt for more details.
- In these cases a screen recording will be wildly helpful.

How to record screen

1. In Chrome Browser go to Google Screen Recorder (https://toolbox.googleapps.com/apps/screen_recorder/)
2. You do not need Audio so when an option is presented block microphone
3. Have your VSCODE files open and the terminal opened and the command typed and ready.
4. Any unwanted applications or windows must be minimized or closed.
5. Hit the “Record” Button in the Screen Recorder Page.
6. Choose “Entire Screen”, Select Scrren and hit “Share”
7. Minimize the browser and your VSCODE with right setup should be in foreground now.
8. Perform your testings and once done go back to the browser and hit “Stop”
9. Choose “Download” and download the recording
10. You can now submit the recording as part of your submission.





How will you be marked

- Working functions in the following order will get points accordingly
  - rolldice() - 1 Point
  - getplayers() - 1 Point
  - getrounds() - 1 Point
  - setgame() - 4 Points
  - asktoroll(player) - 1 Point
  - findwinner(game,players) - 2 Points
  - rungame() - 5 Points
  - printgame() - 5 Points
  - Complete Coverage and Correct flow chart or detailed description of flow chart in words - 3 Points
  - Proper explanation, manual runtime testing, screen recording and screenshots - Bonus 1 Point
  - Successful working program - Bonus 1 Point