INFORMATICS INSTITUTE OF TECHNOLOGY

Individual Coursework

Module Code	: DOC334
Module Title	: Computer Programming
Module Leader	: Nishan Saliya Harankahawa
Assessment Type	: Individual
Issued Date	: 11th November 2024
Hand-in Date	: 29th November 2024 (on or before 23:59)
Weight	: 40%

Student ID :

Student Name :

Centre :

The department is not responsible if an assignment is lost. To cover this eventuality, you are advised to take a copy of the assignment OR to ensure you have the means of re-creating it.

Exceptional Factors Affecting Your Performance:

Students should submit a Mitigating Circumstances Form with evidence, if they miss the submission deadline. This form must be submitted within five working days of the submission date. The Mitigating Circumstances Form is available on the LMS, in the DOC000 - General Information and Policies section.

Assignment Brief

1. **Procedure for Handling Work:**
   - Follow any specific instructions given on the assignment specification.

2. **Penalties for Late Hand In:**
   - If a student submits coursework late but within 24 hours of the specified deadline, the work will be marked but 10% of the overall marks obtained is deducted, to a minimum of the pass mark (40%).

   - If students submit coursework later than 24 hours after the specified deadline, the work is **not** marked and will be given a mark of **zero** for the work in question.

3. **Exceptional Factors Affecting your Performance:**
   - Students should submit written evidence to the Registrar's Department with a copy to the Module Leader of exceptional circumstances, which they consider having caused them to submit assessments late and for which they do not wish to subject for any penalty. The required documents must be handed over to the Registrar within four working days of the hand-in-date.
   - Proper use of Python 3.x coding and language constructs is needed for a better program. You should follow good and proper programming techniques when completing this coursework.

*** Plagiarism ***

The strength of the university depends on academic and personal integrity. You must be honest and truthful. Plagiarism is the use of someone else's work, words, or ideas as if they were your own.

Plagiarism is a serious offense and will not be treated lightly.

Deliverables

The following should be submitted.

- A report including a description of the problem statement and the solution you have developed.
  - Make sure you have followed the good practices of report writing covered by [***DOC311***] Academic Skills for Higher Education module.
    - The report should have standard topics/chapters while below topics must also be added to it.
      - Explanation of the algorithm you used
      - Source code (No screenshots are allowed)
      - Screenshots of the running program
      - Screenshots of the text file generated (display the file content)
      - Test cases

  - Use good word processing skills learned from [***DOC314***] Digital Skills for Higher Education.
  - All program codes **must be included** in your report as **text** (**NOT** image screen captures).
- Test cases used to test the programs and the results must be included in your submission. You can display reasonable amount of test cases.
- You need to provide the **PDF version** of your report and a ZIPPED folder which contains the full and working Python 3.x program
  - Save both (codes and PDF) to a single Zip folder and name it with your IIT student ID number.
  - For example, a zipped folder called **20249999** that carries PDF report and the full final Python program.
- All codes must be written in Python 3.x version
- The completed coursework must be submitted to the LMS via the given link. **DO NOT** email it to your lecturers.
- A **viva session may be arranged** individually to test your knowledge about the CW you submitted or to clarify doubts about your ICW submission.

Scenario

You are to create a **console Python 3.x program** which will allow users to demonstrate a single-player game called "MathsBro".

How it's played

Users will be activating the program by typing the name **mathbro** in the console. This is a simple math game that will generate random maths questions where users will be able to answer. Once all the answers are given by the player, the program will display a result sheet stating the question, user's answer and the status of the answer (correct or incorrect; marked with two symbols √ and X). For every incorrect answer, the program will display the correct answer.

If a user runs the mathbro game just typing the name in the console, the game will be running in demo mode showing only the addition (+) questions where values are ranging from 0 to 5. Users also have the ability to modify the difficulty of the game by passing extra console arguments as below.

No	Console Command	Description
1	`C:\Users\Nishan\mathbro>mathbro`	This is the demo mode. Game will display 3 arithmetic questions with additions (+) only. The operands will range from 0 to 5
2	`C:\Users\Nishan\mathbro>mathbro -e`	This is the easy mode. Game will display 5 arithmetic questions with addition (+) and subtraction (-) only. The operands will range from 0 to 10
3	`C:\Users\Nishan\mathbro>mathbro -m`	This is the medium mode. Game will display 10 arithmetic questions with additions (+)

		and subtraction (-) only. The operands will range from 0 to 10
4	`C:\Users\Nishan\mathbro>mathbro -h`	This is the hard mode. Game will display 10 arithmetic questions with additions (+), subtraction (-) and multiplications (*). The operands will range from 0 to 20

Once the game is started, the users have the option to keep playing the game multiple times with in one session. If the user prefers to exit the game, the session will end. Every game session must be writing to a text file. One game session may have multiple game plays! The text file will carry below sample information.

Sample text files generated are shown below (for the easy mode)

If 1 easy game is played per session

```
File  Edit  Format  View  Help
Date : 2024-11-02
Time : 12:15

Session 1
Result sheet
√  6-6 = 0
√  5+0 = 5
X  1-5 = 63  correct answer is -4
√  7+2 = 8
X  9+5 = 45  correct answer is 14

Total questions :  5
Correct questions :  3
Marks : 60.0%
Level : Easy
```

Ln 15, Col 13 | 100% | Windows (CRLF) | UTF-8

If 2 easy games are played per session

```
File  Edit  Format  View  Help
Date : 2024-11-02
Time : 12:25

Session 1
Result sheet
√  6-6 = 0
√  5+0 = 5
X  1-5 = 63  correct answer is -4
√  7+2 = 8
X  9+5 = 45  correct answer is 14

Total questions :  5
Correct questions :  3
Marks : 60.0%
Level : Easy

Session 2
Result sheet
√  2-2 = 0
√  4+2 = 6
√  8-5 = 3
√  7+5 = 12
X  9+1 = 11  correct answer is 10

Total questions :  5
Correct questions :  4
Marks : 80.0%
Level : Easy
```

Ln 17, Col 10 | 80% | Windows (CRLF) | UTF-8

Your Task

You are to create a Python 3.x program to develop this game. You must create a Python program which runs in the console. You get **zero marks** for providing GUI base answers! You can decide how the game is presented for the end user (the appearance). There is no need to collect usernames, passwords of the player. There is no need for a menu as the game mode is selected via console argument options. You must clearly state your assumptions if you have any.

Tasks to Complete

1. You must use proper Python 3.x program constructs such as packages, modules, functions, variables, data structures, etc. to develop this program.
2. The player must be able to do below tasks
    - View the past game play history
        1. This information is stored by using text files
        2. The extension must be of .TXT
        3. The text file name is a combination of date, time (of saving) and 3 random numbers. For example, if the game is played on 2024-11-15 12:15 PM, the file name will be 20241115_1215_456.txt
    - One game session may have 1 or more game plays
    - Your program should keep track of correct answer count and percentage of each game play.
        1. There can be multiple records for multiple game plays of each session.
        2. The text file is created for the session, not for the game play! (refer the 2 images attached)
        3. There will be multiple game save files for past sessions. Each file must record all the game stats of that session.
        4. The game stats in the text file are
            1. Date of session
            2. Time of session
            3. Which session
            4. Correct/incorrect status of the question
            5. Question
            6. Your answer
            7. Correct answer (if you are wrong)
            8. Total number of questions
            9. Correct number of questions
            10. Marks (correct number of questions as a percentage)
            11. Game mode played
3. You might use external package which will help you to beautify the game interface. However,
    - Such package uses must be **explained** in the report.
    - Proper instructions **must** be given to do the installation of such packages.
    - Links where you install/download the packages **must** also be provided in the report
    - Failing to do above will result lower marks!
    - You **CANNOT** use ready-made game packages for this game! Such attempt will get ***zero marks*** for this ICW.

4. A challenge activity will be to save/display the result in a HTML file so you can see the past game plays in a web browser if you like.
    - You'll get marks for completing this task

End of Coursework
