

PYTHON PROGRAMMING ASSESSMENT 2 LEVEL 1

Objective:

This assignment will test your understanding of file handling, data types, flow control, data structures, and data cleaning in Python. You will create a Python script that processes a text file, extracts specific data, performs calculations, cleans the data, and outputs the results to another file.

Task Overview:

Data Preparation:

You will be provided with a text file named `student_scores.txt`. This file contains student names and their respective scores in three subjects: Mathematics, Python, and Machine Learning. Each line in the file is formatted as follows:

Student_Name, Mathematics_Score, Python_Score, Machine_Learning_Score

Task Requirements:

Step 1: Read and Clean the Data (20 marks)

Read the contents of the file `student_scores.txt`.

Perform data cleaning:

- Remove any leading or trailing whitespace from the names and scores.
- Ensure all scores are valid numbers (handle cases where scores are missing, non-numeric, or out of a typical 0-100 range).
- Convert all scores to integers.
- Impute or ignore any records with invalid data (e.g., if any score is missing or non-numeric after attempts to clean).

Step 2: Process the Data (30 marks)

Calculate the following for each valid student record:

- Total score across all subjects.
- Average score across all subjects.
- Grade based on the average score:

UD: 90 and above (UNDOUBTED DISTINCTION)

DN: 80-89 (DISTINCTION)

MD: 70-79 (MARGINAL DISTINCTION)

D: 60-69 (CREDIT)

P: 50-59 (PASS)

F: < 50 (FAIL)

- Store the results in a list of dictionaries, where each dictionary represents a student. Each dictionary should have the following keys: name, total_score, average_score, and grade.

Step 3: Write the Processed Data to a New File (20 marks)

- Write the cleaned and processed data to a new file named processed_scores.txt.
- The file should be formatted as follows:

StudentName, TotalScore, AverageScore, Grade

Step 4: Document Your Code (10 marks)

- Use comments to document your code, explaining the purpose of each section and any important decisions you made.

Step 5: Code Quality and Modularity (20 marks)

- Ensure your code is well-structured, using functions to handle specific tasks (e.g., reading the file, cleaning the data, processing the data, writing the file).
- Implement error handling for scenarios like file not found, incorrect data format, or invalid data.

Bonus Task:

(5 bonus marks) Implement a feature that handles duplicate student names by appending a unique identifier to each duplicate name (e.g. PreciousMsonda_111, PreciousMsonda_222).

Submission Instructions:

Submit your Python script along with the processed_scores.txt file generated by your code. Make sure your script is well-commented and adheres to Python coding standards.

DEADLINE:

30th August 2024 12:00 NOON

SUBMIT TO:

<https://drive.google.com/drive/folders/1JliTObT2RrCMt9RE7OvqksR1e5KroFJ?usp=sharing>

GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP 5
1. Rahema Likongaji	1. Lingalithu Kadewere	1. Josophat Banda	1. Viwongo China	1. Favour Kampangire
2. Steven Butao	1. Lumbani Kalimamwendo	2. Esther Wanda	2. King Mkandawire	2. EkariOrama Kachepa
3. Olivine Kalisasa	3. James feleston	3. Doreen malanga	3. Zaheera AbdulRashid	3. Brian Nankhuni
4. Maureen Chunga	4. Rodrick Kapoor	4. Gabriel Sherif	4. Thokozani Masina	4. Hannifa Makunganya
5. Veronica khoswe	4. Judith Chawanje	5. Louis mwamsako	5. George Mapelela	5. Thando Mwandira
2. Jonathan Banda	2. Fitina Sichali	5. Mirriam Chiphaka	5. Patricia Sichali	3. Steve Mollen