Assignment 1 – Meteo Data

Assessment Number	1
Contribution to overall mark	30%
Submission Deadline	10/02/2023 at 5:00 pm

Overview

This exercise aims at developing a Python program that handles a kind of small data base system. A sample of csv files is given enabling you to get an idea of the kind the data we are dealing with but also test your code. Your Python code will provide operations for inserting, extracting, deleting and displaying information from the system with appropriate Exception handling.

Problem Statement

Meteorological offices keep records of average monthly rainfalls over a number of cities. The record for each city consists of the name of the city, the year the data refers to, and a list of twelve numbers describing respectively the average rainfall in each of the twelve months of the year. We want to write a Python class system that handles such records. The intended use includes the setting up of a number of such records, and the printing (on request) of available information to the screen. To this end, you need to

Task 1 (25 Marks)

Create a simple class system meeting the stated requirements. Provide a method that calculates the average rainfall over a specified number of months for a given city and a given year.

> ***Hint:*** *Implement the class RainFallRecord and make sure that its constructor carefully validates any data that is to be assigned to the class attributes. Use the data in at least one of the provided csv files to test your code; you should create a Driver class to test your solution.*

Task 2 (30 Marks)

Provide implementations for the following operations (**please stick to the given method signature** (i.e. name, parameter list and return value) for a given year and a given city:

1. rainfall takes as input a month and return the value of the rainfall in the given month of the year, and city. **[5]**
2. delete takes as input a month and deletes the rainfall value associated with the given month, year, and city. **[5]**
3. insert takes as inputs a month and a rainfall value and then inserts the given rainfall value for the given year, month and city. For simplicity, assume that any value that was there before will be lost. **[10]**
4. insert_quater takes as inputs a quarter (winter, spring, summer, autumn) and a list of values and then inserts the given rainfall values for the given quarter and city. Also, assume that any value that was there before will be lost. **[10]**

Hint: *Don't forget that your code must be robust, i.e. return some answer (e.g., an error message) even if the particular operation cannot be completed successfully*. As in Task 1, test your code accordingly.

Task 3 (45 Marks)

Define a class Archive, which will be able to store information about a collection of rainfall records. Provide implementations for the following operations:

1. An appropriate constructor initializing any object of that class. **[10]**
2. A method insert to add a rainfall record to the database. **[10]**
3. A method delete to delete a rainfall record from the database. **[5]**
4. A method sma that takes as inputs a city, a year one, a year two, and a number of months k to return the k months moving averages of rainfall over that city from year one to year two. Find out how to compute the simple moving average. **[20]**

Hint: *Remember that combinations "city name + year" should be unique, no two records in the database should have the same pair of values for the two attributes mentioned above. Again, your code must be robust*. Use the data in at least two of the csv files to test your code.

How to submit

For this assignment, you can work individually or pair with one other student. You should submit the followings:

1. The Python source code written in order to complete the tasks set in the paper. You should submit the Python code file, group1_solution.py or group1_solution.ipynb. Note that even if you decide to work on your own, you must enrol yourself into a group.

2. A signed coursework cover – this should include the names of all the students involved in the work submitted.

Please put your source code and the signed coursework cover into a zip file CW1_GroupID.zip (e.g., CW1_Group1.zip) and then submit your assignment through the module's Blackboard site by the deadline. Note that to submit, you need to click on the Coursework link on Blackboard and then upload your zipped file. Remember that it is **1 submission per group**!

Marking Criteria

For each of the tasks 1 to 3, full marks will be awarded depending on **correctness**, **comments,** and **quality of the code** (e.g., structure, robustness, efficiency); see the associated **marking rubric from Blackboard** for more details.

Marking Group Work

Normally, a group will be given the same mark unless some members made little or no contributions. Any group can be called for an interview during marking. All group members **must attend**, explain their contributions, and defend the work submitted.

Last Updated 14 January 2023 by Emmanuel Tadjouddine