CSM0120
Programming for Scientists

Assignment 02

Weighting: 60%

Release date: Thursday, 9th November 2023

Deadline: Thursday, 11th January 2024 at 1:00pm

Type: Blackboard Submission

Yasir Saleem

yss1@aber.ac.uk

1. Submission and Assessment

This assignment is worth **60%** of the marks for CSM0120. You should submit a single zip file containing your code and your report. The report should be in PDF format, with contents as described below in Section 2.2.

The zip file is to be submitted via Blackboard before 1:00pm on Thursday, 11th January 2024. The department takes late submission of assignments very seriously and any work submitted after the deadline will receive zero marks unless an extension has been agreed before submission.

By submitting via Blackboard, you are implicitly declaring the work to be your own.

The body of the report must be in your own words; it is not acceptable to construct your report by copying and pasting chunks of text from the web or any other source without appropriate attribution[1]. It is important to indicate clearly in your own work where you have included the work of others. In Computer Science, this could include reuse of designs or code, as well as copying text, or use of generative AI.

Marking will be anonymous and will be according to the assessment criteria for Development, Appendix AA of the student handbook[2]. Please do not include any identification information.

Feedback will be returned on or before Thursday, 1st February 2024.

In case of personal, financial or health problems affecting this coursework, please provide a special circumstances form[3] to your year coordinator (for Master's students taking Computer Science modules, this is Myra Wilson, mxw@aber.ac.uk).

If you have specific questions relating to the assignment itself, please contact Yasir Saleem, yss1@aber.ac.uk.

2. The Assignment

This assignment is comprised of two parts: programming and report. In programming, you are expected to write pieces of code for several tasks mentioned in Section 2.1. In the report, you are expected to provide details about four aspects mentioned in Section 2.2.

[1] Please take note of the statement on Plagiarism and Unfair Practice which can be found in Department Handbook that can be found under DEPT-N-PG in Blackboard of the University Regulation on unfair practice (https://www.aber.ac.uk/en/academic-registry/handbook/regulations/uap/)

[2] https://impacs-inter.dcs.aber.ac.uk/images/editor-content/Documentation/Handbooks/Appendices/AppendixAA.pdf

[3] https://www.aber.ac.uk/en/academic-registry/handbook/taught-schemes/name-193260-en.html

2.1 Programming

Code quality: Make sure all the code you are going to write will be available as reusable functions: at least one function for each question you have answered. Your functions should take parameters/arguments and should return values. Add a main function which demonstrates how to use your other functions. Your code should be well commented, well tested, well documented, with proper exception handling and should be able to identify incorrect input. Your code should be written elegantly and should produce the correct output. The marks for each task (mentioned below) are also associated to code quality.

Part 1: UK Map

Task 1: [10 marks] Plotting Towns in England and Wales on UK Map

You are provided a file called `latlon.csv` that contains latitudes and longitudes for towns in England and Wales. Each line contains 3 fields: `town name, latitude and longitude`. You are also given a module `UKMap.py`, which provides a class `UKMap` with handy functions (note that an image `Gb4dot_merged_mapcolors.png`, provided on Blackboard with assignment 2 resources, is required to run `UKMap.py`).

Write a Python function `plot_specific_towns` to read in the towns data (from `latlon.csv`) and plot those towns on a `UKMap` which fulfil any of the following criteria.

- The town name starts with A, B, C, L or M.
- The town name ends with “bury” or “ampton”.

You must make sure that Aberystwyth, Birmingham, Cardiff and London are easily found on the map, i.e., choose a different marker and marker size for them or perhaps annotate them. Test your code and demonstrate that you are confident that it is producing a correct plot. Due to the fact that the Earth is spherical and maps are two-dimensional, there will be some distortion when plotting locations. For instance, you may find that some locations fall slightly below the south coast.

Part 2: Weather

This part of assignment requires you to develop programs which automatically access weather forecasts of the next five days of 9 cities in England and Wales, and provide general suggestions to the residents in these cities. The 9 cities considered in this assignment include:

Aberystwyth	*Bangor*	*Birmingham*
Cardiff	*Derby*	*Leeds*
London	*Manchester*	*Swansea*

In this part, you are asked to perform the following tasks:

- Automatically access online data;
- Extract, process and store key information from online data;
- Present the key information in XML form elegantly;
- Plot the cities on UK Map indicating different weather conditions and indicating which cities have more subscribed users than users.

You will be using the free weather API from `WeatherAPI` to get access to the weather forecast data. The link to `WeatherAPI` is https://www.weatherapi.com. In particular, you will be using the **14 day weather forecast** API from `WeatherAPI`. Its documentation is available at https://www.weatherapi.com/docs/. You should fetch the temperature in **Celcius** for both APIs.

You need to first register with `WeatherAPI` before using its API. Please use the free option while registering on the website. Once you complete the registration, you will be able to get your **API key** by clicking My Account.

This API key is a combination of 31 alphanumeric characters in a similar form as follow:

2edb**********2805

Please provide your API key in the submission so that your program can work properly.

When you will use **14 day weather forecast** API from `WeatherAPI`, we are mainly interested in **time (datetime)**, **temp_c (temperature)** and **condition->text (weather condition)** attributes of hourly forecast. You can find them under **`forecast->forecastday->hour->time`** (for datetime), **`forecast->forecastday->hour->temp_c`** (for temperature) and **`forecast->forecastday->hour->condition->text`** (for weather condition) for each hour.

Task 2: [10 marks] Accessing weather data using API

Write a program that can access the hourly weather forecast of 3 days at 9 cities (mentioned above) using `WeatherAPI`. The weather forecast data should cover 72 hours (i.e., 24 hours x 3 days) for each city with a forecasting interval of one hour.

Task 3: [15 marks] Extracting and storing information from online data

Write a program that processes weather forecast data obtained from `WeatherAPI` (obtained in Task 1) so that only the information of datetime, weather condition and temperature is stored in csv files. The program should store the extracted forecast of 3 days at 9 cities into 9 individual csv files (one csv file per city) in a similar form as mentioned in the example below in Table 1.

For each csv file, name it as *<city name>.csv*. For example, if the csv file stores the information extract at Aberystwyth, name the csv file as `Aberystwyth.csv`.

Table 1. Example for format of storing data into csv files.

Date and time	*Weather condition*	*Temperature*
2023-11-02 00:00	*Moderate rain*	*7.4*
2023-11-02 01:00	*Light rain*	*8.0*
2023-11-02 02:00	*Light rain*	*8.7*
…	*…*	*…*
2023-11-04 23:00	*Mist*	*7.2*

Task 4: [20 marks] Summarizing key information from online data

Write a program that divides the 9 cities into four categories: `raining`, `snowing`, `icing` and `else` based on the weather condition and temperature in the next day (e.g., if today is 2023-11-02, the program should only consider the weather forecast data covering the period from 2023-11-03 00:00:00 to 2023-11-03 23:59:59) and prints suggestion(s) for the citizens in the console following the template given below.

At a time, a city can appear in one of the four categories based on the following criteria:

- A city will be categorised as `raining` if the value of weather condition (`condition->text`) contains `"rain"` or `"drizzle"` for at least six hours between the time period from 06:00:00 to 23:59:59.
- Otherwise, a city will be categorized as `snowing` if the value of weather condition (`condition->text`) contains `"snow"` or `"blizzard"` for at least four hours between the time period from 00:00:00 to 23:59:59.
- Otherwise, a city will be categorized as `icing` if the temperature is below 0 Celsius degree for more than eight hours between the time period from 00:00:00 to 23:59:59.
- Otherwise, a city belongs to the category `else`.

Based on the categories that the 9 cities belong to, the program should print a suggestion message on the console for the citizens living in the 9 cities. For the cities belonging to the category `raining`, citizens should receive a suggestion of “Bring your umbrella”. For the cities belonging to the category `icing`, citizens should receive a suggestion of “Mind your step” as water on the floor may get frozen. For the cities belonging to the category `snowing`, citizens should receive a suggestion of “Plan your journey thoroughly”. For the cities belonging to the category `else`, citizens should receive a suggestion of “Enjoy the weather”. The template of the message is given as follows (“XXXXXXXXXX” is a city name).

Enjoy the weather if you are living in these cities:
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
Bring your umbrella if you are in these cities:
XXXXXXXXXX

XXXXXXXXXX
XXXXXXXXXX
Mind your step if you are in these cities:
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
Plan your journey thoroughly if you are in these cities:
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX

Note that, if there is no city belonging to a specify category, there is no need for giving that suggestion. For example, if all the cities are going to rain in the next day, the message should appear as:

Bring your umbrella if you are in these cities:
Aberystwyth
Bangor
Birmingham
Cardiff
Derby
Leeds
London
Manchester
Swansea

If four cities belong to the category `else` and the other five cities belong to the category `icing`, the message should appear as:

Enjoy the weather if you are in these cities:
Aberystwyth
Bangor
Birmingham
Manchester

Mind your step if you are in these cities:
Cardiff
Derby
Leeds
London
Swansea

Task 5: [15 marks] Presenting data in the XML format

Write a program that converts the message (from Task 4) into an easy-to-interpret XML form and save as *<date>.xml*. The purpose is that other people can directly access the general suggestions without executing the program. Your XML data should follow the template provided below in Listing 1.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<WeatherForcasting>
   <Date Date="2023-11-03">
      <GoodWeather>
         Enjoy the weather if you are in these cities
         <cities>
            <city name="XXXXXXXXXXX" />
            <city name="XXXXXXXXXXX" />
            <city name="XXXXXXXXXXX" />
         </cities>
      </GoodWeather>
      <PoorWeather Issue="Raining">
         Bring your umbrella if you are in these cities
         <cities>
            <city name="XXXXXXXXXXX" />
            <city name="XXXXXXXXXXX" />
            <city name="XXXXXXXXXXX" />
         </cities>
      </PoorWeather>
      <PoorWeather Issue="Icing">
         Mind your step if you are in these cities
         <cities>
            <city name="XXXXXXXXXXX" />
            <city name="XXXXXXXXXXX" />
            <city name="XXXXXXXXXXX" />
         </cities>
      </PoorWeather>
      <PoorWeather Issue="Snowing">
         Plan your journey thoroughly if you are in these cities
         <cities>
            <city name="XXXXXXXXXXX" />
            <city name="XXXXXXXXXXX" />
            <city name="XXXXXXXXXXX" />
         </cities>
      </PoorWeather>
   </Date>
</WeatherForcasting>
```

Listing 1. XML Format of printing summarized forecast temperature information.

Task 6: [10 marks] Plotting Cities Showing Weather Conditions and Size of Subscribed Users

You are provided with a file `users.csv` that shows the users in the UK who have signed up to receive weather information. Write a program that plots all 9 cities from `latlon.csv` on UK map using `UKMap.py` (as done in Task 1). This time, the marker size of each city should reflect the number of users who have signed up for that city (you need to count the number of subscribers for each city using `users.csv`) and the choice of marker shape should be as follows:

- Use red triangle if it is raining
- Use cyan diamond if it is icing
- Use blue star if it is snowing
- Use green circle if it is anything else

Note that if no user is subscribed to a city, then you should not plot anything for that city on the map.

2.2 Report [20 marks]

Write a report about your work. The report will have four sections given equal weight:

i. Executive summary

Give an outline of the work done. This section should describe what your program does without reference to implementation details and should be easy to understand for someone who has never programmed before.

ii. Technical overview

Provide justification of the implementation choices you made. This section should explain, for example, what kinds of loops you have used and why, what kinds of data structures you have used, how you have dealt with any special or unusual cases, how you processed the data and how you produced the output.

iii. Software testing

Describe how you can be sure that the output you produce is correct. Describe how you have tested the individual functions and how you tested the code as a whole.

iv. Reflections and future work

Discuss how your code could be improved and extended. Include an estimate of the total time you took on this exercise.

The report should be submitted in PDF form only. As a guideline, it should contain around 3500 words. You can include diagrams and screenshots if these are helpful. A good report will be clear, concise and well-illustrated.

Note: Overly long reports (>3850 words) and reports submitted in formats other than PDF will be penalised.