CSM0120 Assessed Coursework 2

Deadline: Tuesday January 9th 2018 at 4pm (Blackboard submission)

1 Submission and Assessment

This assignment is worth **60%** of the marks for CSM0120. You should submit a **single zip file** containing your code, your database and your report. The report must be in PDF format, with contents as described below. The zip file is to be submitted via Blackboard before 4pm on January 9th 2018.

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. Please take note of the information on Unacceptable Academic Practice which can be found in the module DEPT-N-PG on Blackboard. 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 and code as well as copying or quoting text.

Marking will be anonymous and will be according to the assessment criteria for Development, Appendix AA of the taught postgraduate handbook (also in DEPT-N-PG).

Feedback will be returned on or before January 30th 2018.

In case of personal, financial or health problems affecting this coursework, please provide a special circumstances form `https://www.aber.ac.uk/en/aqro/exams/special-circumstances/` to your year coordinator (for Masters students taking computer science modules this is Edel Sherratt, eds@aber.ac.uk).

If you have specific questions relating to the assignment itself, please contact Amanda Clare, afc@aber.ac.uk.

2 The Assignment

The assignment involves the following components

1. **[10 marks]** You are given a file called `latlon.csv` containing latitudes and longitudes for towns in England and Wales. Each line contains 3 fields: town name, latitude, longitude. Have a look at this file. You are also given a module UKMap.py. Have a look at this file. It provides a class `UKMap` with handy functions. Write a Python function `plot_all_towns` to read in the towns data and plot it on a UKMap. 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 2-dimensional, there will be some distortion when plotting locations (for example you may find that some locations fall slightly below the south coast).

2. **[10 marks]** You are providing a personalised weather service to a set of users who have signed up to your service. The data about the users who have signed up is provided in the file `users.csv`. Have a look at this file. Design a database schema that will hold the data from the `users.csv` file and also the data from the `latlon.csv` file. Provide a diagram of your database schema in your report. Create your database. Write Python code to enter the data from both files into your database. You might find that some users have towns that are not present in the latlon data. Report any problem towns and make a decision what to do about them. Test your code and demonstrate that you are confident that all data from both files has been added to the database.

3. **[10 marks]** Read the documentation about how to use the `locationforecast` service from `http://api.met.no/#english` at
`http://api.met.no/weatherapi/locationforecast/1.9/documentation`

Using the `requests` and `BeautifulSoup` libraries (which we will be dicussing in lectures) write a function `forecast`, that:

- downloads a forecast for a particular location,
- saves the resulting XML text response as a suitably named file
- returns an `BeautifulSoup` object representing the XML data

Aberystwyth has latitude 52.41616 and longitude -4.064598. Use your function to request the weather forecast for Aberystwyth.

4. **[10 marks]** Some of the `location` elements of the forecast contain a `windSpeed` element. The `location` elements are found within a `time` element that has `from` and `to` attributes. Write a function `wind_forecast` to extract all the times and wind details from a given forecast `Element` and make a suitable plot showing how the wind speed will vary for the week ahead. The times represent an interval over which each prediction holds. You will have to make a reasoned decision about how best to use this information. Make sure that your graph is well labelled. Inspect some of the values to ensure they look correct.

5. **[20 marks]** Write a function to do the following: For a given email address, look up the user's location in your database, get the corresponding forecast, extract only the very first `windSpeed` and `windDirection` values from the forecast and plot them on the UKMap. Use a triangle marker, rotated by the wind direction, and coloured to show the wind speed. Use the 5 colours `["blue", "green", "yellow", "orange", "red"]` or use a more detailed colour palette of your choice. A triangle marker can be created by the tuple `(3, 0, deg)` where `deg` is the degrees of rotation that you would like to use. Be careful to save all downloaded XML as sensibly named files so that you can reproduce your data if asked, because all scientists need to be able to reproduce their results.

6. **[20 marks]** Be creative in representing any other interesting weather forecasts for groups of your users using your data (and, if you want to be really creative, any other data whose use is not restricted).

7. **[20 marks]** Write a report about your work. The report will have the following four sections, equally weighted at 5 marks each:

   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's never programmed before.

   Technical overview Justify the implementation choices you made. This section should explain, for example, your database design, what kinds of loops you've used and why, what kinds of data structures you've used, how you've dealt with any problem cases, how you processed the data and how you produced the output.

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

   Reflections and future work Discuss how your code could be improved and extended.

   The report should be submitted in PDF form only. As a guideline, it should contain around 4000 words. You can include diagrams and screenshots if these are helpful. There is no limit on diagrams and screenshots. The word limit is short, leaving no room for waffle. A good report will be clear, concise and well-illustrated. Note: Overly long reports (>4500 words) will be penalised, as will reports submitted in formats other than PDF. The report should be included in your single zip file together with the code and any additional files you would like (e.g. plots) for the submission.