# 4COM1037 – 2016/2017 Semester A

# Programming Task Part 1 – Completed as Coursework

You have been contacted by the Met Office to write a program to display city weather around the world. They will supply the data and you are to design, code and test part of a simple prototype system with the functionality detailed below.

This system will form the basis for a test under exam conditions during which you will be asked to modify and extend it. You must therefore ensure that your design conforms to the specification below. You must develop and submit your system in the single .groovy file that you extract from the downloaded list of .groovy files on Studynet. The file has your name and id. The Groovy code must work with the file located online as documented below and in the code. This work itself is not marked but the demo (test) will be, so you need to complete it yourself and understand the code so you can modify it. Please write your last-name, first-name and SRN number at the top of your script as comment. **This must be completed as an individual piece of work – each student works independently. Do not share your work with others.**

The table below provides the **required** behaviour for this prototype system.

|  |  |
| --- | --- |
|  | **Comment** |
| ***Data*** | **The system will load data from a file.** |
| **File of weather data** | You are supplied with a file and an online link to a copy of the file. You may use either file but be aware that you must not change the format because the online file will be used for marking your demo. There is also a method that loads the file and converts it into a 2D array. If you are confident, you may change the conversion if you like as long as you can work with your code in the demo. Please see the Groovy code for further documentation.  The file is formatted with every value on a new line in groups of five as below:  *City name*  *Today’s high (temperature in Celsius)*  *Today’s low (temperature in Celsius)*  *Rainfall in millimetres*  *Today’s pressure (hPa)*  So, the first group (the first city’s data) could be:  *London*  *20.5*  *14.1*  *2*  *1014*  The data would be updated every day but the format would stay the same. |
| **Behaviour** | **(Functional Requirements)** |
| *Start-up behaviour* | Your system should print this welcome message:  “Welcome to Weather View” |
| *Menu* | Your system must have a text menu that allows a user to do the following:   * Print a list of city names * Print a list of all cities and their weather data * Search for a city * Show the city with the highest temperature * Exit the system gracefully   The exact format of the menu is your decision as is the value that a User would supply to choose a given item and to exit the system. |
| *User input* | You must make use of JOptionPane to get user input |
| *Print a list of cities* | Just the city names should be printed, one on each line |
| *Print a list of all cities and their weather data* | This should print out the name of the city and its weather data in a nicely formatted way. |
| *Search for a city* | The User supplies the name of a required city and the system searches the array and either prints out the city and its weather data or a message saying that it was not found. |
| *Show the city with highest temperature* | Searches the array and finds the city with the highest high temperature that day. Prints out the city name and the temperature. |
| *Exit the system* | The user exits the system and some message is displayed to show that the program has finished running. |
| *General Requirements (non functional requirements)* | The program should be designed and coded using techniques that you have been shown in class. DO NOT go online and get Groovy code or Groovy solutions because they may not be easily modified or extended during the in-class demo.  Your code should be subdivided into functions and be written so that it is easily maintainable and extendable. You do not need to worry about error handling in the event of a user making a mistake or deliberately inputting incorrect information. So for example, if your program requires a user to enter an int and they supply a word or character you are not required to provide any error handling as we have not learned this. You may if you wish as long as you can work with the code during the demo. |