

Programming Project 5 - Data Analysis

Note: When you turn in an assignment to be graded in this class you are making the claim that you neither gave nor received assistance on the work you turned in (except, of course, assistance from the instructor).

Program Name: DataAnalysis.java

You will write a program that will accept a text file containing the high temperature, low temperature and rainfall amount for each day of a given month of the year for a selected city. The first line has the year, month and city separated by tabs. The following lines contain the data separated by tabs. The data includes:

- the high temperature (int),
- the low temperature (int), and
- the rainfall amount (double).

A sample text file is available in this submission.

The program should loop so that the program can read and process more than one data file as needed by the user.

The Java program will read in the month and year to determine the number of days in the month that the data was collected. You will use at least one separate method for this procedure and the method header will be, **private static int numDaysInMonth (String month, int year)**.

The program will read in the temperatures and rainfall amounts into three different arrays: two int arrays for high and low temperature and one double array for rainfall. The program will then calculate the average high temperature and average low temperature with no decimal places for the month. The program will also calculate the total rainfall for the month to two decimal places. You will use a method for both actions.

- One header will be **private static int calculateAverage(int [] temp)**.
- The other array method will be **private static double calculateSum(double [] rainfall)**.

The program will write the results to a file called: **dataOutput.txt**. The first line of output will write the month, year and city name. You will then write average high temperature, average low temperature and total rainfall for the given month, year and city, each on a separate line.

The program will then write to the file the data for each day of the month. This output will start with the given day, followed by the high temperature, low temperature and rainfall amount separated by ample whitespace. In addition, you will also write the following:

- For every day that is above the average, display a "+" beside the temperature. For every day that is below the average, display a "—" beside the temperature.
- For every rainfall amount greater than 0.00, display an "*" beside the rainfall amount.
- Numbers should line up so that the ones digits are all in the same column.

The program will then ask the user if they want to read another text file of weather information. If yes, then the program will loop back up to read the new file and execute again.

Save your program in a file called **DataAnalysis.java**. Output for the sample file:

For the month of September 2017 in Richmond:
The average high temperature was 81 degrees.
The average low temperature was 61 degrees.
The total rainfall was 1.48 inches.

```
1  76- 53- 0.83*
2  67- 57- 0.41*
3  80- 62+ 0.06*
4  83+ 56- 0.00
5  86+ 64+ 0.00
6  74- 61 0.07*
7  76- 54- 0.00
8  78- 51- 0.00
9  74- 55- 0.00
10 71- 51- 0.00
11 72- 53- 0.02*
12 79- 60- 0.03*
13 84+ 65+ 0.06*
14 83+ 65+ 0.00
15 85+ 67+ 0.00
16 86+ 65+ 0.00
17 85+ 64+ 0.00
18 81 66+ 0.00
19 84+ 67+ 0.00
20 89+ 66+ 0.00
21 90+ 65+ 0.00
22 84+ 62+ 0.00
23 87+ 61 0.00
24 90+ 62+ 0.00
25 88+ 64+ 0.00
26 78- 69+ 0.00
27 90+ 72+ 0.00
28 88+ 62+ 0.00
29 78- 55- 0.00
30 72- 51- 0.00
```

Reminder: this program and all programming projects in this course must include a comment block at the beginning of the source code file that contains:

- your name
- course and semester information
- project number
- a brief description of what the program does
- instructions for the user

In addition to your rocket information, the program **must display** specific identifying information grouped together in a block of code in a header method within the program. Make sure you call the header method within the main method of your code. In the code block, write Java statements that **output to the console** the following:

- your name
- the course number and section
- the Java program name
- the project number

Fully document your algorithm or each action within your code using comment blocks.

Document your tests using the form shown below.

You will submit your Java source code file (DataAnalysis.java), and test documentation (TestPlan.docx) by uploading the files to the Assignment link in Blackboard. Please do not submit your files in a zipped folder. Also, your test plan must be a word or PDF file.

For extra credit, you can fully handle all exceptions that might arise when running your file.

Ask questions about any part of the programming project that is not clear!

Test Plan:

Expected output – expected test results against which the output of the test is compared.

Test Inputs	Expected Output	Actual Output
2017 September	30 Days	30 Days
2100 February	28 Days	28 Days
2004 February	29 Days	29 Days
56, 67, 54, 60, 49	57 low temp avg	57 low temp avg
88, 81, 80, 92, 95	87 high temp avg	87 high temp avg
Richmond	Richmond	Richmond
Newport News	Newport News	Newport News
0.03, 0.05, 0.02	0.1 total rainfall	0.1 total rainfall
0.00, 0.00, 0.00	2 decimal places	2 decimal places
value > average	+ next to value	+ next to value
value < average	- next to value	- next to value
rainfall on that day	* next to rainfall	* next to rainfall
bad file name input	prompt again	prompt again
2017 December	31 Days	31 Days
avg temp round up	rounded up	rounded up

Rubric for Programming Project 5

Criteria	Points
Program file named and submitted as specified	5
Comments used appropriately (including comment header information and fully documented algorithm)	15
Appropriate choice of variable names	5
Opens and reads input file correctly	15
Separate methods are used as specified and using good program design	25
Output is correct	10
Program layout and appearance (Coding style is clear and easily understood)	5
Output is written correctly to the output file	10
Test plan completed and submitted with source code	10
TOTAL	100