ACADEMIC YEAR: 2024-2025

ASSIGNMENT 3

Assignment Posted: June 16, 2025

Assignment Due: June 27, 2025, before 11.59 PM

"Clojure enables you to focus on the 'what' (the problem you want to solve), not the 'how' (the low-level details of state and control flow)."

— Rich Hickey

Description:

This assignment offers you a deep dive into the world of Clojure. Although any previous experience with the language isn't required, it presents a valuable chance to explore and apply the key concepts of Clojure taught in class. In addition to reinforcing your knowledge of familiar constructs, you'll encounter new ideas and features that will enrich your hands-on experience and enhance your overall programming skills. You'll engage with Clojure's versatile data structures, such as maps, vectors, sets, and regular expressions. By the end of this task, you'll have developed a stronger command of Clojure and its capabilities.

NOTE: Please note that the assignment document may seem lengthy, but the description of the assignment itself is concise and clear. The document primarily consists of valuable hints and advice designed to streamline development of your Clojure program.

Welcome to the Functional Weather Report System assignment! Get ready to explore the functional programming paradigm by developing a command-line tool that acts as a personal weather report assistant. Weather impacts our daily lives and long-term planning, and in this assignment, you'll take a functional approach to organizing and interacting with weather data.

Weather reports provide us with insights into atmospheric conditions, helping us prepare for everything from daily commutes to planning vacations. As a functional programmer, your task is to create a dynamic collection of weather reports using Clojure. You'll create a text-based interface that enables users to: View, filter, as well as analyze weather data, transform temperature units, work with immutable collections and declarative logic. All logic should be expressed using pure functions, immutable data, and Clojure idioms.

Through this assignment, you will gain experience in designing user-friendly text menus, collecting and transforming input, applying functional operations over collections, and optionally persisting and retrieving data from files.

Your application must begin by displaying a welcome message and a main menu like the one illustrated in Figure 1. It should offer five core options, assuming perfect user input (correct data types and expected formats).

```
=== Weather Report System ===
1. View Weather Reports
2. Transform Weather Report
3. Filter Weather Reports
4. Weather Statistics
5. Save and Exit
Enter your choice (1-5):
```

Figure 1. Main menu

Implementation Specifications:

Detailed behavior of each menu option is mentioned below with screenshots.

1. View Weather Reports

This option enables users to view all stored weather reports. These reports should be read from a user-provided text file and displayed in a neatly formatted layout. Each report must include the date, location, temperature, and condition (e.g., Rainy, Sunny). The total number of weather reports must be displayed at the top, followed by the report details in order of their dates. A sample of the format is given below.

```
=== Weather Report System ===
1. View Weather Reports
2. Transform Weather Report
3. Filter Weather Reports
4. Weather Statistics
5. Save and Exit
Enter your choice (1-5): 1

Total weather reports: 20
-----------------------------------------------------------
    Date     |   Location   | Temperature  |  Condition
-----------------------------------------------------------
 2024-01-01  |    London    |     26°C     |    Cloudy
 2024-01-02  |   New York   |     4°C      |    Rainy
 2024-01-03  |    Tokyo     |     17°C     |    Foggy
 2024-01-04  |    Paris     |     34°C     |    Rainy
 2024-01-05  |    Berlin    |     30°C     |    Stormy
 2024-01-06  |    Sydney    |     20°C     |    Stormy
 2024-01-07  |   Toronto    |     11°C     |    Rainy
 2024-01-08  |    Dubai     |     7°C      |    Cloudy
 2024-01-09  |     Rome     |     32°C     |    Snowy
 2024-01-10  |    Delhi     |     10°C     |    Cloudy
 2024-01-11  |   Beijing    |     2°C      |    Stormy
 2024-01-12  |    Moscow    |     19°C     |    Foggy
 2024-01-13  |     Rio      |     -1°C     |    Stormy
 2024-01-14  |    Madrid    |     -2°C     |    Cloudy
 2024-01-15  |   Bangkok    |     8°C      |    Snowy
 2024-01-16  |   Chicago    |     -5°C     |    Rainy
 2024-01-17  |   Istanbul   |     1°C      |    Sunny
 2024-01-18  |    Seoul     |     -1°C     |    Cloudy
 2024-01-19  | Mexico City  |     23°C     |    Windy
 2024-01-20  |  Cape Town   |     12°C     |    Stormy

=== Weather Report System ===
1. View Weather Reports
2. Transform Weather Report
3. Filter Weather Reports
4. Weather Statistics
5. Save and Exit
```

Figure 2. Menu option 1 “View Weather Reports”

2. Transform Weather Reports

This option enables the users to functionally transform the existing weather data, reinforcing Clojure's emphasis on pure, composable functions and immutable data structures. Upon selecting this option, the user is prompted to choose one of two transformation operations to apply to the dataset. Default unit in the text file is ^{0}C.

- **Convert temperatures Celsius to Fahrenheit:** This transformation traverses the entire collection of reports and converts the temperature value in each map from Celsius to Fahrenheit. It uses the formula F = (C × 9/5) + 32 and demonstrates the application of map to modify specific fields in a data structure.
- **Convert temperatures Fahrenheit to Celsius:** This transformation traverses the entire collection of reports and converts the temperature value in each map from Fahrenheit to Celsius. It uses the formula C = (F-32) × 5/9) and demonstrates the application of map to modify specific fields in a data structure.

```
=== Weather Report System ===
1. View Weather Reports
2. Transform Weather Report
3. Filter Weather Reports
4. Weather Statistics
5. Save and Exit
Enter your choice (1-5): 2
Choose a transformation:
1. Convert temperatures to Fahrenheit
2. Convert temperatures to Celsius
Enter your choice (1-2): 1

Transformed Weather Reports:

Total weather reports: 20
-----------------------------------------------------------
    Date     |    Location    | Temperature |  Condition
-----------------------------------------------------------
 2024-01-01  |     London     |     79°F    |    Cloudy
 2024-01-02  |    New York    |     39°F    |    Rainy
 2024-01-03  |     Tokyo      |     63°F    |    Foggy
 2024-01-04  |     Paris      |     93°F    |    Rainy
 2024-01-05  |     Berlin     |     86°F    |    Stormy
 2024-01-06  |     Sydney     |     68°F    |    Stormy
 2024-01-07  |    Toronto     |     52°F    |    Rainy
 2024-01-08  |     Dubai      |     45°F    |    Cloudy
 2024-01-09  |      Rome      |     90°F    |    Snowy
 2024-01-10  |     Delhi      |     50°F    |    Cloudy
 2024-01-11  |    Beijing     |     36°F    |    Stormy
 2024-01-12  |     Moscow     |     66°F    |    Foggy
 2024-01-13  |      Rio       |     30°F    |    Stormy
 2024-01-14  |     Madrid     |     28°F    |    Cloudy
 2024-01-15  |    Bangkok     |     46°F    |    Snowy
 2024-01-16  |    Chicago     |     23°F    |    Rainy
 2024-01-17  |    Istanbul    |     34°F    |    Sunny
 2024-01-18  |     Seoul      |     30°F    |    Cloudy
 2024-01-19  |  Mexico City   |     73°F    |    Windy
 2024-01-20  |   Cape Town    |     54°F    |    Stormy
```

(a) Celsius to Fahrenheit

```
=== Weather Report System ===
1. View Weather Reports
2. Transform Weather Report
3. Filter Weather Reports
4. Weather Statistics
5. Save and Exit
Enter your choice (1-5): 2
Choose a transformation:
1. Convert temperatures to Fahrenheit
2. Convert temperatures to Celsius
Enter your choice (1-2): 2

Transformed Weather Reports:

Total weather reports: 20
-----------------------------------------------------------
    Date     |    Location    | Temperature |  Condition
-----------------------------------------------------------
 2024-01-01  |     London     |     26°C    |    Cloudy
 2024-01-02  |    New York    |     4°C     |    Rainy
 2024-01-03  |     Tokyo      |     17°C    |    Foggy
 2024-01-04  |     Paris      |     34°C    |    Rainy
 2024-01-05  |     Berlin     |     30°C    |    Stormy
 2024-01-06  |     Sydney     |     20°C    |    Stormy
 2024-01-07  |    Toronto     |     11°C    |    Rainy
 2024-01-08  |     Dubai      |     7°C     |    Cloudy
 2024-01-09  |      Rome      |     32°C    |    Snowy
 2024-01-10  |     Delhi      |     10°C    |    Cloudy
 2024-01-11  |    Beijing     |     2°C     |    Stormy
 2024-01-12  |     Moscow     |     19°C    |    Foggy
 2024-01-13  |      Rio       |     -1°C    |    Stormy
 2024-01-14  |     Madrid     |     -2°C    |    Cloudy
 2024-01-15  |    Bangkok     |     8°C     |    Snowy
 2024-01-16  |    Chicago     |     -5°C    |    Rainy
 2024-01-17  |    Istanbul    |     1°C     |    Sunny
 2024-01-18  |     Seoul      |     -1°C    |    Cloudy
 2024-01-19  |  Mexico City   |     23°C    |    Windy
 2024-01-20  |   Cape Town    |     12°C    |    Stormy
```

(b) Fahrenheit to Celsius

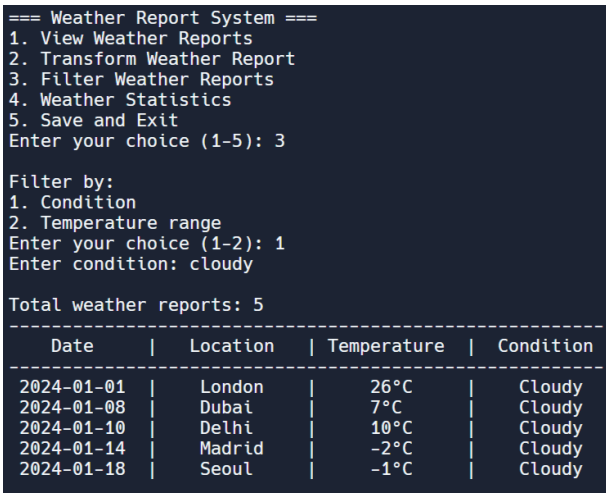
Figure 3. Transform Weather Reports

3. Filter Weather Reports

This option empowers users to selectively explore weather reports based on specific filtering criteria. It improves data usability by allowing users to focus only on the relevant weather conditions or temperature ranges. When selected, the user is presented with a sub-menu to choose from two filtering options:

- **Filter by Condition:** The user is prompted to enter a specific condition, such as "Sunny", “Rainy”, "Cloudy", *etc.* The system displays all reports where condition exactly matches the input. The comparison is case-sensitive.

- **Filter by Temperature Range:** The user is asked to enter minimum and maximum temperature values. The program shows only those reports where the temperature falls within the given range (inclusive).

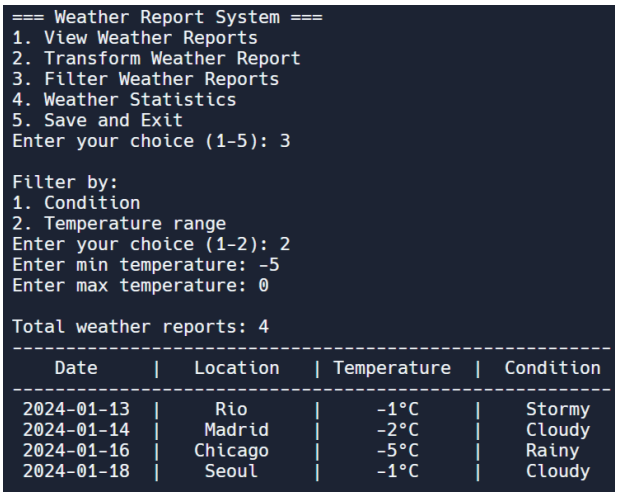


```
=== Weather Report System ===
1. View Weather Reports
2. Transform Weather Report
3. Filter Weather Reports
4. Weather Statistics
5. Save and Exit
Enter your choice (1-5): 3

Filter by:
1. Condition
2. Temperature range
Enter your choice (1-2): 1
Enter condition: cloudy

Total weather reports: 5
------------------------------------------------------------
    Date     |   Location   | Temperature  |  Condition
------------------------------------------------------------
 2024-01-01  |    London    |     26°C     |    Cloudy
 2024-01-08  |    Dubai     |     7°C      |    Cloudy
 2024-01-10  |    Delhi     |     10°C     |    Cloudy
 2024-01-14  |    Madrid    |     -2°C     |    Cloudy
 2024-01-18  |    Seoul     |     -1°C     |    Cloudy
```

(a) Filter by condition



```
=== Weather Report System ===
1. View Weather Reports
2. Transform Weather Report
3. Filter Weather Reports
4. Weather Statistics
5. Save and Exit
Enter your choice (1-5): 3

Filter by:
1. Condition
2. Temperature range
Enter your choice (1-2): 2
Enter min temperature: -5
Enter max temperature: 0

Total weather reports: 4
------------------------------------------------------------
    Date     |   Location   | Temperature  |  Condition
------------------------------------------------------------
 2024-01-13  |     Rio      |     -1°C     |    Stormy
 2024-01-14  |    Madrid    |     -2°C     |    Cloudy
 2024-01-16  |   Chicago    |     -5°C     |    Rainy
 2024-01-18  |    Seoul     |     -1°C     |    Cloudy
```

(b) Filter by Temperature range

Figure 4. Filter Weather Reports

4. Weather Statistics

This option enables users to analyze stored weather reports by generating insightful statistics that summarize the overall data trends. When this feature is selected from the main menu, the program processes the list of weather reports and computes:

- **Average Temperature:** The system calculates mean of all recorded temperatures. This offers a general sense of the climate trends in the data.
- **Hottest and Coldest Days:** The program identifies the reports with the maximum and minimum temperatures respectively, and displays their complete details (date, location, temperature, and condition). The user is asked to enter minimum and maximum temperature values. The program will show only the reports where the temperature falls within the given range (inclusive).
- **Unique Weather Conditions**: Using Clojure's set functionality, the application determines how many distinct weather conditions are present in the dataset (e.g., Sunny, Rainy, Cloudy).

5. Save and Exit

Selecting this option enables users to exit the Weather Report System program. When this option is chosen, the program displays a message indicating that the user is exiting the program. After displaying the exit message, the program terminates, and the user returns to the command line or terminal prompt. This option provides a convenient way for users to save the changes to file and gracefully exit the program after they have completed their tasks or no longer wish to continue using it. ***Saving the changes is optional and this menu option can be treated as exit only.***

```
=== Weather Report System ===
1. View Weather Reports
2. Transform Weather Report
3. Filter Weather Reports
4. Weather Statistics
5. Save and Exit
Enter your choice (1-5): 4

Average temperature: 12.35
Hottest day:  2024-01-04  |    Paris     |     34°C     |    Rainy
Coldest day:  2024-01-16  |   Chicago    |     -5°C     |    Rainy
Unique conditions: 7
```

Figure 5. Weather Statistics

Submission:

The assignment must be submitted only through Moodle. No other form of submission will be considered. Please create a zip file containing your Clojure code and a readme file (txt). The zip file should be named Assignment#3_YourStudentID. In the readme document, the features and functionality of the application, and anything else you want grader to know.

IMPORTANT – FUNCTIONAL PROGRAMMING CONSTRAINTS: Your entire application must be implemented using **purely functional constructs and idioms**, as idiomatic to the Clojure programming language. You must avoid all forms of imperative or state-mutating logic such as loop, while, or mutable data structures (e.g., atoms, refs, agents, or vars used for mutation). Instead, you must use **immutable data**, **first-class functions**, and **higher-order functions** such as map, filter, reduce, group-by, partition, sort-by, etc., to express all logic in a **declarative** and **composable** manner. Input-output should be confined to clearly defined boundary functions, and all transformation logic should be expressed as side-effect-free pure functions. Submissions that rely on imperative constructs or violate functional purity will receive significant penalties, even if the output appears correct. The goal of this assignment is to demonstrate your understanding of **functional design patterns** and your ability to structure an application **entirely without explicit looping or state mutation**.

Additional Information

- You can use the code provided during the tutorial or lecture sessions to get started.

Evaluation Procedure

Please note that the markers will be using a standard Linux system, likely the same docker installation that most of you should be using. Your code MUST compile and run from the Linux command line. If you are using an IDE (e.g., IntelliJ or VS Code), rather than a simple editor, you will want to test execution from the command line, since IDEs often create their own paths, folders, environment variables that might prevent the code from running on the command line.

To evaluate the submissions, the marker will simply create a test folder and add a sample input data file (weather_data.txt) plus your source files. Your code will be compiled and run and each of the menu options will be tested. Every student will be graded the same way. The markers will be given a simple spreadsheet that lists the criteria described above.

There are no mystery requirements. Please note that it is better to have a working version of a slightly restricted program than a non-working version of something that tries to do everything (e.g., you might be able to display the entries, but you can't format it, or you can display the existing entries, but you can't add any). Hence, make sure that your code compiles and runs. It is virtually impossible to grade an assignment that does not run.

THIS ASSIGNMENT MUST BE SUBMITTED INDIVIDUALLY.

MANDATORY DATA STRUCTURES AND USAGE:

1. **Maps:** Each weather report is a map with keys: :date, :location, :temperature, :unit, and :condition.
2. **Vectors:** All weather reports are stored in a vector. Use vector operations for filtering and iteration.
3. **Sets:** Use sets to compute the number of distinct weather conditions.
4. **Filtering and Sorting:** Use filter, map, reduce, sort-by, group-by, for transformation and querying.

IMPORTANT!!!

Markers will receive a straightforward spreadsheet containing the specified criteria outlined earlier. There are no undisclosed or hidden requirements. It is emphasized that having a functional version of a somewhat limited program is preferable to having a non-functional version attempting to cover every aspect. Therefore, prioritize ensuring that your code can be successfully compiled and executed. Evaluating an assignment that doesn't run at all is virtually impossible.