Lab 8.5: Reading and Processing Monthly Temperatures

Purpose

This lab introduces the use of **structs** to store related data, file I/O for reading external data, and arrays for managing multiple records. Students will also learn to work with **GitHub for version control**, making meaningful commits as they complete the lab.

By the end of this lab, you will:

1. Define and use a **struct** to store a **day's temperature**.
2. Read **temperature data from a file** into an **array of structs**.
3. Print the temperature data in a formatted table.
4. Compute the **average temperature**.
5. Use **GitHub** to track progress, making incremental commits as you complete different parts of the lab.

Procedure

Pre-Lab Reading Assignment

1. Before starting the lab, read the section on **structs** in your textbook or lecture notes. Pay close attention to how structs can be used to group related data.
2. Register for GitHub and accept the invitation in the Canvas assignment.

Lab Setup

1. Clone the GitHub Repository:
    a) Accept the GitHub Classroom invitation link provided by your instructor.
    b) Clone the GitHub Repository in Visual Studio 2019/2022:

        i. Accept the GitHub Classroom invitation link provided by your instructor.
        ii. Open Visual Studio 2019/2022.
        iii. Click on **Clone a repository** or in the program, go to **File > Clone Repository**.
        iv. Paste the GitHub repository URL and choose a local directory and create a new folder and select it.
            1) Click the …
            2) Select a folder
            3) Create a new subfolder
        v. Click **Clone** to download the repository.
        vi. Once cloned, create a new project as you have done previously.

1) **File > New Project**
2) Choose **Empty Project**
3) Click **Next**
4) Enter a name and the same location as your cloned repository. You can find the subfolder under your created folder as shown: NewFolder\. Select it.
5) Check "Place solution and project in the same directory." Click **Create**
6) In the Solution Explorer, add the source files as you have previously done by right-clicking on Source Files and Add Existing Items. You should be able to navigate to the root folder that has the code file you will need (array_of_structs.cpp). Choose it and click add.

vii. Ensure you have the file `temps.txt` in the same directory as your project file.

2. **Examine `temps.txt`**:
   i. This file contains daily temperature readings for a month.
   ii. Each line contains a day number followed by a temperature:

```
1 68
2 70
3 72
...
30 61
```

Lab Assignment

Step 1: Define the Struct *(Estimated Time: 10-15 minutes)*

1. Create a struct named `TemperatureRecord` that has two integer fields:

   - `day` (stores the day number: 1-31)
   - `temperature` (stores the temperature in degrees Fahrenheit)

2. Declare an array of `TemperatureRecord` structs
3. Modify the code to declare this struct in the global section.
4. Commit to GitHub: `"Created TemperatureRecord struct to store day and temperature"`
   a) After defining the struct and array, go to the 'Team Explorer' window in Visual Studio:
      i. Click 'View' in the top menu, then select 'Team Explorer' (or press Ctrl+\, Ctrl+M).
   b) In Team Explorer, click the 'Git Changes' tab (it may say 'Pending Changes').
   c) You'll see your modified file (e.g., main.cpp) listed under 'Changes'. You may need to scroll down. Select it.
   d) Click the + sign on the line next to the modified array_of_structs.cpp file (the file you changed). Save changes when prompted.

e) In the 'Commit' section above the list of files, type the commit message: "Created TemperatureRecord struct to store day and temperature".
f) Click Commit Staged

The next step may not be necessary (if grayed out).

g) After committing, click the 'Sync' link  to open the 'Synchronization' view.

h) In the 'Outgoing Commits' section, click 'Push'  to send your commit to the GitHub repository.
   i. If prompted, sign in with your GitHub credentials.
   ii. Check your GitHub repo online to confirm the commit appears.

Step 2: Read Data into an Array of Structs *(Estimated Time: 15-20 minutes)*

1. Modify the `readTemperatures` function to:
   a) Open `temps.txt`.
   b) Read the day and temperature into an array of `TemperatureRecord` structs.
      - Use a while loop to read two integers per line (day and temperature) into the array.
      - For each line, store the values in the next available struct in the array. Hint: Think about how to access the 'day' and 'temperature' fields of a struct using the array index. (e.g. array[index].structParameter, where structParameter represents what you named the day and temperature variables in the struct).
      - Track the number of records read using the size parameter (passed by reference).
   c) Ensure no more than 31 records are stored.
2. Commit to GitHub: `"Implemented readTemperatures function to load data into array of structs"` using the previous steps.

Step 3: Print Data *(Estimated Time: 10-15 minutes)*

1. Implement `printTemperatures` to print all stored temperatures in a formatted table:

   Example Output:

```
Day  Temp
1    68
2    70
...
30   61
```

2. Commit to GitHub: `"Implemented printTemperatures function"` using the previous steps.

Step 4: Compute the Average Temperature *(Estimated Time: 15-20 minutes)*

1. Implement `findAverage` to calculate and return the average temperature.
2. Commit to GitHub: `"Implemented findAverage function"` using the previous steps.

Final Commit & Submission

1. Ensure all functions are working.
2. Fix any issues.
3. Push final changes to GitHub: `"Formatted output and finalized program"` using the previous steps.
