Assignment 1 Journal: Basic Programming Problems

Boris Bojanov
ID:3608550

Date Started: Sep 8 2024
Date Completed: Sep 25 2024

Initial Thoughts and Planning

The first assignment involved solving five basic programming problems in C++. These included creating multiplication tables, developing an interactive help menu, and identifying prime numbers up to 10,000. Each problem required understanding of loops, conditionals, input/output formatting, and nested structures. I anticipated that creating robust test plans for each solution would require significant effort.

I created an action plan for each problem, breaking down requirements and key tasks. For Problem 1, I needed to create a multiplication table from 1× to 12×, considering different formatting options for numbers of various lengths. I aimed to solve each problem incrementally and maintained organized, documented code from the start.

Problem 1: Multiplication Table from 1× to 12×

Process and Challenges

I used nested `for` loops to generate the multiplication values. Proper formatting was key to ensuring columns aligned regardless of the number of digits. I used `setw()` from `<iomanip>` for consistent spacing, which improved readability.

Reflections

This problem reinforced my understanding of nested loops and output formatting. I enjoyed the visual results and immediate feedback from the table output.

Problem 2: Temperature Conversion Tool

Process and Challenges

Problem 2 involved creating a tool to convert between Fahrenheit and Celsius. I wrote the conversion formulas as functions and handled both uppercase and lowercase inputs using `tolower()`. Adding error handling ensured the program could prompt users to re-enter invalid inputs.

Reflections

This problem was engaging due to user interaction. Designing helpful error messages and handling unexpected inputs improved my understanding of creating a user-friendly interface.

Problem 3: Temperature Conversion Table

Process and Challenges

Problem 3 required static methods for temperature conversions and formatted output. Using `fixed` and `setprecision(3)` solved decimal formatting issues. Representing both positive and negative values in a clean table format was rewarding.

Reflections

The use of `static` methods was new for me, and it was a good learning experience. I also gained confidence in advanced formatting techniques.

Problem 4: Interactive C++ Help System

Process and Challenges

For Problem 4, I created an interactive help system using a menu and a `switch` statement. Researching definitions and writing concise explanations took time, but it was crucial to making the help system useful.

Reflections

This problem emphasized the importance of user experience. Ensuring the help menu was easy to navigate helped me understand the value of intuitive UI design.

Problem 5: Identifying Prime Numbers from 1 to 10,000

Process and Challenges

I used nested `for` loops to identify prime numbers, optimizing by iterating up to the square root of each number. This reduced runtime significantly, and experimenting with a `bool` flag helped mark non-prime numbers efficiently.

Reflections

This problem pushed me to optimize my approach, which was a valuable exercise in algorithm efficiency.

Summary Reflections

Assignment 1 covered different aspects of C++ programming, including user input, output formatting, and algorithm optimization. It encouraged me to think critically about design decisions, such as using static methods and optimizing prime-checking algorithms. Keeping this journal helped me organize my thoughts, document challenges, and reflect on my progress.

I look forward to the next assignment and the opportunity to deepen my understanding of more complex programming challenges.

Sources

https://devdocs.io/cpp/