

Assignment 3 Journal: Object-Oriented Problems

Boris Bojanov

ID: 3608550

Date Started: Oct 30 2024

Date Completed: Nov 27 2024

Initial Thoughts and Planning

Assignment 3 focused on C++ file I/O and array manipulation. The six problems involved working with files, creating classes, managing arrays, and implementing a custom Set template. I anticipated gaining a stronger grasp of file handling, data structures, and template programming.

I began by mapping out each problem's requirements and key tasks. For Problems 1-3, I needed to work extensively with file operations and input handling, while Problems 4-6 involved array and class manipulation. This plan helped me efficiently manage time and track progress.

Problem 1: Counting Words in a File

Process and Challenges

For Problem 1, I wrote a program that opens a file and counts the number of whitespace-separated words. I added error handling for file access and tested the solution using various text files, including the provided `excerpt.txt`. I chose to obtain the file name via prompt, as it seemed more user-friendly than command-line arguments.

Reflections

This problem enhanced my understanding of file input in C++, particularly reading files line-by-line and processing content efficiently.

Problem 2: Display File Contents One Line at a Time

Process and Challenges

In Problem 2, I wrote a program that displays file contents one line at a time, prompting the user to press **<Enter>** for each new line. Handling the pause between lines required using `cin.get()` to ensure the correct behavior.

Reflections

The challenge of keeping user interaction smooth was interesting. It allowed me to focus on user experience when working with file input.

Problem 3: TextFileReader Class

Process and Challenges

I created a `TextFileReader` class that used an array to store file lines, with two constructors and functions to display contents. The `display()` function added line numbers to each output. I tested using `TextFileReaderDemo`, which handled command-line arguments.

Reflections

This problem highlighted the use of constructors and multiple methods for file handling within a class. It reinforced the value of organizing related operations within a class for cleaner design.

Problem 4: Floating Point Array Operations

Process and Challenges

Problem 4 involved creating three floating-point arrays. The first two were filled with loop counter values and their squares, while the third stored the sum of corresponding elements. I used nested loops to display the arrays in the required format.

Reflections

This problem emphasized array operations and effective use of loops for calculations. It was a straightforward task that reinforced my understanding of array manipulation.

Problem 5: Book and Bookshelf Classes

Process and Challenges

I extended the **Book** class from Assignment 2, adding new attributes and creating a **Bookshelf** class to manage multiple books. I implemented sorting using a custom comparator and displayed book attributes before and after sorting.

Reflections

This problem provided valuable practice with data encapsulation and sorting logic, especially using comparators to manage complex sorting conditions.

Problem 6: Custom Set Template Class

Process and Challenges

I implemented a **Set** template using a C++ Standard Library vector for internal storage. The challenge was ensuring unique elements and creating an iterator class for traversal. Testing involved comparing behavior with the standard C++ set.

Reflections

This problem introduced me to creating template classes and iterators. It demonstrated how using STL components can simplify custom data structures while retaining unique functionality.

Summary Reflections

Assignment 3 provided extensive practice with file handling, arrays, and template programming. Each problem had unique aspects that required different C++ skills, from managing file I/O to creating robust data structures. Documenting my progress helped me understand the importance of well-planned and structured coding practices.

Sources

<https://devdocs.io/cpp/>