Scientific Programming

Assignment: Array - Operator Overloading.

Anand Kamble

amk23j@fsu.edu

24th October 2023

The Array class implementation is a dynamic array structure in C++. In this Array class each element is a 2x2 matrix, and it also provides methods for multiplication, printing, modification of these matrices.

Code Modules

The code is organized in the following modules:

- `Array.h/.cpp` : Array class definitions and methods, which allow us to manipulate the dynamic array.
- `main.cpp`: The main function which implements the Array class and tests its methods.
- `Test.cpp` : The 'Test' class which is used to test the any values with expected values.

Formatting Conventions:

a. Casing: Camel-Case
b. Indentation: 4 spaces
c. Line Break: CRLF

Array Class

Constructors and Destructor:

- Default Constructor (Array()): Initializes an empty array.
- Parameterized Constructor (Array(int size)): Initializes the array with a specified size. [1]
- Copy Constructor (Array(const Array &)): Enables the creation of a new object by deep copying an existing one.
- Destructor (~Array()): Frees allocated memory upon the object's destruction.
- Assignment Operator (Array &operator=(const Array &)): Overloaded assignment operator for copying the contents of one Array` object to another.

Array Operations:

Element-wise Multiplication (`Array &operator*(const Array &)): Overloaded multiplication operator for multiplying two arrays element-wise.

Scalar Multiplication (friend Array &operator*(float, Array &)): Overloaded multiplication operator for multiplying an array by a scalar.

Element-wise Sum of Multiplication (`int operator%(const Array &)): Overloaded modulo operator for calculating the sum of element-wise multiplication between two arrays.

Element Access (`int* operator[]): Overloaded subscript operator for accessing elements of the array by index.

Push, Pop, Remove, Insert Operations: Methods for dynamically manipulating the size and content of the array.

Printing (`void print()): Outputs the elements of the array to the console.

Execution

The Makefile is included with this code. You can run the command 'make' to compile the program. After successful compilation, you can find the executable named 'test.x' inside the bin folder. Run this executable by './bin/test.x'. To clean the generated folders and files, use the command 'make clean'.