**D1166570 Levi**

# Programming Assignment 5 Report:

# Complex Number Operator Overloading

## Introduction:

Embarking on the journey of implementing complex number operator overloading in C++ provided an enriching experience into the realms of object-oriented programming and operator overloading. This report outlines the steps undertaken, challenges faced, and insights gained throughout this assignment, which required defining a ‘Complex’ class to perform arithmetic operations and solve quadratic equations.

## Dynamic Memory Management:

A core aspect of this assignment involved efficient memory management. In C++, dynamic memory allocation is crucial for handling objects that are created and destroyed during runtime. The ‘Complex’ class, though simple in structure, required careful consideration of memory handling, particularly when dealing with complex numbers during arithmetic operations.

## Implementing Complex Number Operations:

The primary task was to implement fundamental arithmetic operations for complex numbers: addition, subtraction, multiplication, division, and calculating the absolute value. Each of these operations was overloaded using member functions and friend functions to ensure they could be used intuitively with the Complex class.

2024/5/15

# Dealing with Edge Cases:

Handling edge cases such as division by zero and operations with zero-valued complex numbers was critical. Implementing checks within the overloaded operators ensured the robustness of the class, preventing erroneous operations and maintaining integrity under various conditions.

# Testing the Implementation:

To validate the functionality of the Complex class, comprehensive testing was conducted. This included:

- Performing a series of arithmetic operations.
- Solving quadratic equations and verifying roots.
- Ensuring precision up to six decimal places, as specified in the assignment.

# Organizing the Code:

Adopting a modular approach, the code was organized into separate files:

- Complex.h and Complex.cpp for the Complex class definition and implementation.
- quadratic_equation_verifier.cpp for the application program to solve quadratic equations.

This structure enhanced maintainability and readability, facilitating easier debugging and future enhancements.

# Reflections and Conclusions:

This project was a profound exercise in understanding the intricacies of operator overloading and object-oriented programming in C++. It highlighted the power of C++ in managing complex data structures and performing sophisticated operations with ease. Each step of the implementation deepened my

appreciation for the language and refined my programming skills.

In conclusion, this assignment was not just an academic requirement but a significant learning experience. It provided valuable insights into advanced programming concepts and prepared me for more challenging endeavors in the future. As I move forward, I carry with me the lessons learned and a renewed curiosity to explore further into the realms of computer science.

The following includes the result of my assignment.