Report_5_D1265154 曾郁珊

1. Differences in Programming Complex Number Assignments Using C and C++

C Programming:

In my experience with C programming, handling complex numbers often involves defining structures or utilizing libraries like complex.h. I typically define custom structures to represent complex numbers, which include separate fields for the real and imaginary parts. Operations on complex numbers are usually implemented using functions, resulting in a procedural approach to coding.

For instance, in C, I commonly define a structure for complex numbers like so:

```
typedef struct {
    double real;
    double imag;
} Complex;
```

Then, I implement functions to perform various operations such as addition, subtraction, multiplication, and division.

C++ Programming:

Transitioning to C++, I've found that the language offers built-in support for complex numbers through the <complex> library. Alternatively, I can define my own Complex class, as demonstrated in the provided code snippet. In C++, complex numbers can be manipulated using operator overloading and encapsulation within classes, resulting in a more intuitive and object-oriented approach to complex number handling.

2. Advantages and Disadvantages of Programming in C++

Advantages of Programming in C++:

- Namespace Management: In C++ facilitate namespace management by encapsulating related complex number functionality within a single namespace. This prevents naming conflicts and organizes code logically, improving code readability and scalability in the Complex program.
- Encapsulation: Classes allow me to encapsulate the data (real and imaginary parts) and operations (addition, subtraction, etc.) related to complex numbers within a single entity. This encapsulation promotes data integrity by restricting direct access to internal members and providing controlled interfaces for interacting with complex numbers.

Disadvantages of Programming in C++:

- Difficulty in Testing: The encapsulation of data and functionality within classes can make unit testing more challenging in the Complex program. Testing individual class components may require intricate setups or mocking of dependencies, especially when dealing with complex interactions between classes. This complexity can hinder the effectiveness of unit testing and increase the likelihood of undiscovered defects.

- Memory Management: Manual memory management in C++, particularly using features like new and delete, poses challenges in the Complex program. Improper memory allocation and deallocation can lead to memory leaks, impacting program performance and reliability.