

Report of assignment 5

Jensen D1265209

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

In C programming, complex number assignments and operations require the use of structures and functions because C does not support operator overloading or classes. To handle complex numbers, we need to define a structure that contains the real and imaginary parts, and write functions to perform complex number assignments and operations. For example, we define a structure to represent a complex number and write functions to implement addition, subtraction, multiplication, and division of complex numbers.

C++ allows the use of classes and operator overloading, making it easier to work with complex numbers. In C++, we can define a class to encapsulate the real and imaginary parts of a complex number, and use operator overloading to implement addition, subtraction, multiplication, and division operations. This allows us to directly use operators to manipulate complex numbers, making the syntax more intuitive and readable. Additionally, C++ supports encapsulation, enabling complex numbers' data and methods to be organized within a class, providing better data protection and structure.

The main differences are: In C, we use structures (struct) to represent complex numbers, while in C++, we use classes (class) to encapsulate complex numbers and their related operations. In C, we need to write functions to perform complex number operations, whereas in C++, we can use operator overloading, making complex number operations more intuitive. C++ supports encapsulation, allowing complex numbers' data and methods to be organized within a class, providing better data protection and structure.

2. Advantages and Disadvantages of Programming in C++

Programming in C++ has several advantages. Firstly, C++ supports object-oriented programming (OOP), which facilitates better data abstraction, encapsulation, inheritance, and polymorphism, promoting code reuse and modularity. Secondly,

C++ allows operator overloading, making it more intuitive to manipulate custom data types, such as complex numbers, resulting in more readable and maintainable code. Additionally, C++'s Standard Template Library (STL) provides a rich set of template classes and functions for common data structures and algorithms, reducing the need for custom implementations. C++ also offers robust error handling mechanisms through exceptions, enabling cleaner error management and recovery. Lastly, C++ provides both static and dynamic memory management, giving developers control over resource allocation and deallocation.

However, C++ also has some disadvantages. Firstly, compared to C, C++ is a more complex language with a steeper learning curve due to its extensive feature set. Secondly, features like exception handling, Run-Time Type Information (RTTI), and virtual functions can introduce performance overhead. Furthermore, object-oriented programming can sometimes lead to more verbose code compared to procedural programming, especially for simple tasks. Lastly, while C++ is backward compatible with C, mixing C and C++ code can result in complex and hard-to-maintain codebases.