Assignment Report for Complex Number Programming in C and in C++

Corrine Tseng

1. **Differences in Programming Complex Number Assignments Using C and C++**
    - **(1) Complex numbers in C**
        - **1) Standard library support**
            First of all, C does not have built-in support for complex numbers in the C89 standard. Additionally, The C99 standard introduced complex numbers with the <complex.h> header, which defines the complex type and functions for complex arithmetic.
        - **2) Operations**
            Basic operations such as addition, subtraction, multiplication, and division are performed using predefined functions.
        - **3) Limitations**
            Lack of operator overloading means that functions must be used for all operations. Less intuitive and harder to read compared to C++.
    - **(2) Complex numbers in C++**
        - **1) Standard library support**
            C++ provides a <complex> header that includes a std::complex template class, offering extensive support for complex number operations.
        - **2) Operations**
            C++ allows operator overloading, making complex number arithmetic more intuitive and readable.
        - **3) User-defined complex class**
            C++ allows the creation of user-defined classes with operator overloading.
2. **Advantages and disadvantages of programming in C++**
    - **(1) Advantages of programming in C++**
        - **1) Object-Oriented Programming (OOP)**
            C++ supports OOP, which allows for encapsulation, inheritance, and polymorphism. OOP improves code modularity and reusability.
        - **2) Operator Overloading**
            Allows for more natural and intuitive expression of arithmetic operations. Enhances readability and maintainability of code.
        - **3) Performance**
            C++ allows for low-level memory manipulation similar to C, offering high performance. Inline functions and templates can optimize performance by avoiding function call overhead.
    - **(2) Disadvantages of programming in C++**
        - **1) Complexity**
            The language complexity and extensive feature set can lead to steep learning curves for beginners. More complex syntax and semantics compared to simpler languages like Python.
        - **2) Undefined Behavior**
            Greater risk of undefined behavior due to pointers, manual memory management, and other low-level features. Requires careful programming and understanding of language rules.
        - **3) Verbosity**
            More verbose than some higher-level languages, which can lead to larger codebases.