Assignment 5 Report

A real number and an imaginary number are two floating-point numbers that are frequently used to represent complex numbers in computer programming. Though their dual nature makes their mathematical operations a little more difficult, these numbers may be worked with and altered just like any other sort of data. In order to show the distinctions between C and C++ and to address the benefits and drawbacks of C++ programming, this report will examine how complex number operations are implemented in both languages. In C programming, complex number operations are typically implemented using structures and functions. There is no built-in support for complex numbers, so developers must define their own data structure to represent complex numbers and then implement functions to perform arithmetic operations on them. In contrast, C++ offers native support for complex numbers through the <complex> header, which defines a templated class std::complex. This class encapsulates the real and imaginary parts of a complex number and provides

overloaded operators for arithmetic operations. In this semester, transitioning from C to C++ opened some challenges. With C++, we delve into object-oriented programming (OOP), where we can encapsulate data and behavior into classes, offering a more organized approach to programming. Operator overloading lets us define custom behaviors for operators like '+', '-', '*', and '/', making code more expressive and readable. Templates enable writing generic code, which can be reused across different data types to save time.