Lab report 3:

1.

a.

In this lab assignment, the concepts I explored were related to classes to create and implement a ComplexNumbers class, overloading operators to add,subtract,multiply and divide complex numbers. Other concepts used for these objectives are object oriented programming, using constructors, getters and setters, and debugging. These concepts are very important as it allows code to be more manageable and readable. When working as a software developer, I will be collaborating with a lot of individuals, thus making sure my code is readable and manageable by breaking it down into classes is very helpful and satisfies the industrial standards.Moreover, these classes can be reused and there is no need to write them again and again.
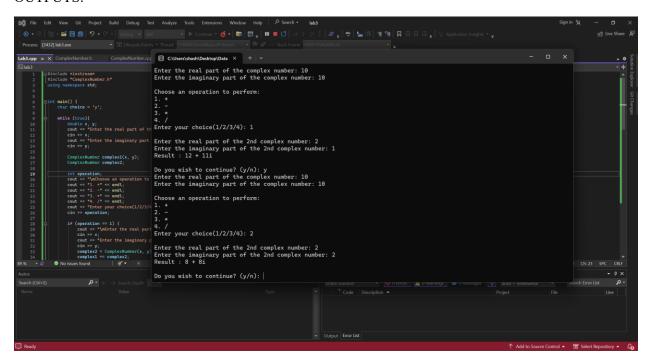
b.

Initially, I designed the class to have 2 private members to store the real and imaginary parts of a complex number. Next, I added overloaded constructors, getters and setters to access the private members of the class. And in the end I added the print member function to display complex number output. The changes made included that the +,- operators accepted input of same class type and returned void, before that it returned output of same classtype. I had used the class type to take input and return the same type for all operations so I went back and changed it to void in the header file and declaration file. Another mistake I made was using complex1 = complex1 + complex2 instead of using the declared operator complex1 += complex2. I was able to fix this issue by debugging which took most of the time to find and caused a lot of build errors.
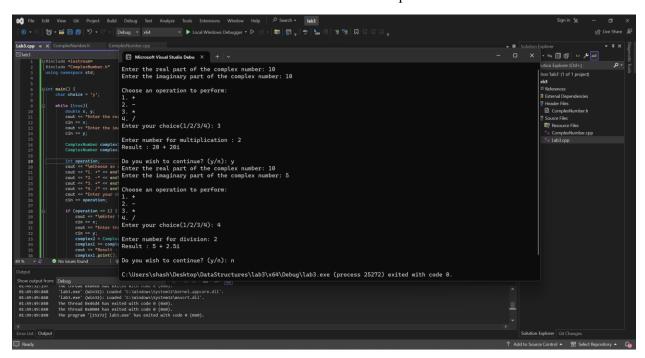
2.

Instructions to compile the program : Make sure all the files are in the same directory. The main file is Lab3.cpp, the header file is ComplexNumber.h, and the definitions file is ComplexNumber.cpp. Make sure the file name of header file is the same as it has been included as "#include "ComplexNumber.h"". Also the source code has been attached in the zip folder submitted. Please refer to that.
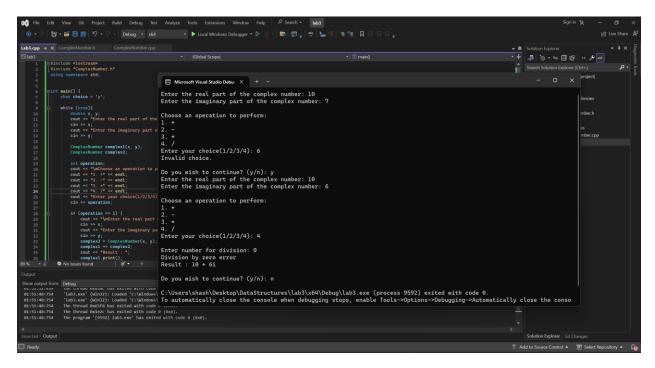
OUTPUTS:

Addition and subtraction output

Multiplication and division output

Division by zero and invalid input error test cases