Jacob Kennedy, Henry Stiefel, Alex Turner

Data Structures (CS 2028C)

Lab 3

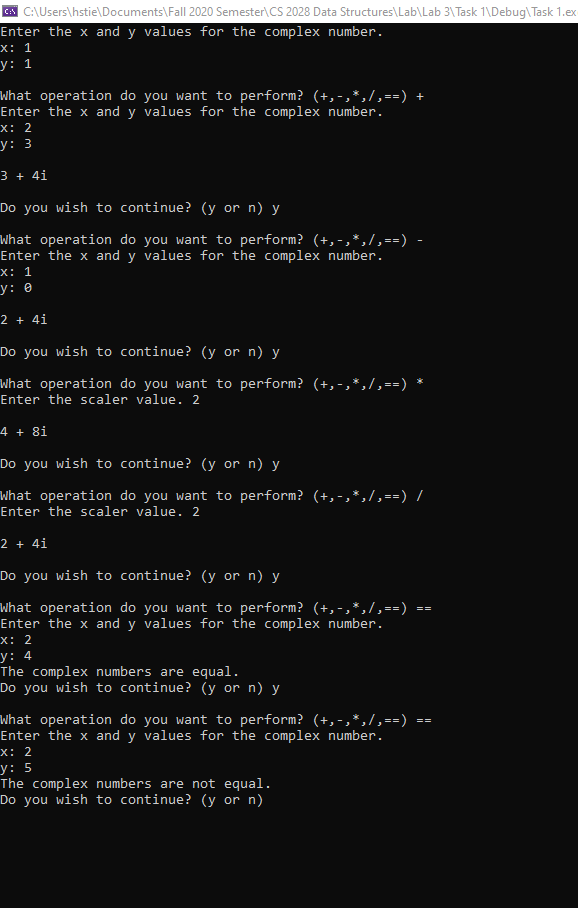
Rationale for Class Members

The members used in my ComplexNumber class are two double variables, myX and myY, and a bool called isCartesian. The doubles represent the x and y components of a complex number, and they can be used to calculate the phi angle or r value for the polar form. They are doubles to allow for values with decimal points. The bool, isCartesian, simply keeps track of whether or not the instance of ComplexNumber is representing Cartesian form or not. There are three constructor members – a default constructor, accepting no arguments, a constructor accepting two double argument, and a third constructor accepting two doubles and a bool value. The constructor with three arguments allows the user to explicitly declare the ComplexNumber Cartesian or not using the bool parameter. There are getter and setter methods for the private member variables myX and myY. There are two additional getter methods for the r value and angle phi value, necessary for outputting a ComplexNumber in polar form. Next, there is the printValue function and the five overloaded operator methods.

Changes Made to Class Declaration

Overall, there were no changes needed in the declaration file during task 2. During task 3, we realized that we did not need to store r and phi as member variables, as they could be calculated using just x and y. Also, since the x and y values are subject to change, using getter methods instead of member variables means the r and phi values will never be outdated (for example, the case that an x or y value changes but r or phi isn’t properly updated.)

Screen shot showing the output of all operator overload functions.



1. The objectives and concepts explored in this assignment are the creation and implementation of a class, the overloading of constructors, and the overloading of operators. Class creation is very important because it allows us to create blueprints which can be used to manage objects and keep track of data. Classes bundle data together in the form of member variables and allow us to manipulate and modify that data with member functions. Knowing how to write and implement our own custom classes is crucial for solving problems using computer science techniques. Constructor overloading is important because it allows us to use the same class for differing representations. For example, the two constructors used in this lab allowed us to create a complex number in one of two forms, Cartesian or Polar. This is the concept of polymorphism, one of the four pillars of object-oriented programming. Finally, operator overloading is important because it allows us to perform common mathematical operations on the member variables in our custom classes.
2. Initially, the class held four member variables for x, y, r, and phi. As we worked on task 3, however, we realized r and phi could be accessed with a getter method using calculations based on x and y, and it would be best practice to remove them as member variables. Another change we made was with our constructors. There is the default constructor, an overloaded constructor that takes two parameters, and an overloaded constructor that takes three parameters. We initially thought the one with three parameters should use three doubles, for x, y, and phi, however we realized since phi is calculated using x and y that it can’t be a part of the constructor. We changed this constructor to take a bool value which determines whether the number is in Cartesian or Polar form. One of the most important considerations to make when designing a class is deciding the scope of variables. We could have x and y public, however this could lead to memory issues and data being changed when not intended, so using public getter and setter methods is the best way to implement member variables.