Complex Number Multiplication. (CNM)

Define a Struct that has two member variables of data type double.

The Struct will have a member function that will take an output and format the member variables as ordered pairs (which can represent complex numbers on the complex plane).

The CNM program will overload the multiplication \* operator (Changing what the operator does while used in this program) to perform the multiplication of complex numbers, this will look like FOIL of a binomial expression, anytime the operator is used.

Three functions will be declared as the type of the struct and will take two instances of the struct as parameters.

One function will display two complex numbers that are declared in the program, the member function will format the output as (num, num) ordered pairs.

Another function will take the same objects as parameters, multiply them to be performed as

(a1, b1) \*(a2, b2) again, distributive property or FOIL.

and display the product using the output from the member function. The function returns a struct object of the product.

A third function will take two uninitialized Struct objects as parameters, they will hold ordered pairs entered by the user which are then multiplied, and the product will be displayed again using the output from the member function. This one also returns a struct object of the product.

The main function will initialize two struct objects with values of the ordered pairs and declare two uninitialized Struct objects that will hold user input. The three functions will all be called from here.

Here is an example out intended I/O:

// Complex Number Multiplier

// Point F is (3,5)

// Point G is (7,2)

// The Product of F & G is (prod, prod)

// Enter numbers for two ordered pairs represented as points V & C

// Enter x for V: A

// Please enter a number.

// Enter x for V: 2

// Enter y for V: 3

// Enter x for C: 2

// Enter y for C: 2

// Point V is (2,3) Point C is (2,2)

// The product of point V & C is (prod, prod)