Question 1 : Use operator overloading as member function in C++ to compute the sum and difference of the points and then compute the distance between point-1 and point-2 using Pythagorean formula.

Question 2 : Use operator overloading as non-member function in C++ to compute the sum and difference of the points and then compute the distance between point-1 and point-2 using Pythagorean formula.

Note: You can use the math library for square root computation.
The solution of both the questions should have different files for implementation and interfaces e.g., consists of a header file, a .cpp and a driver class (which having the main function).
<u>The desired output in both cases will be as follows:</u>
Point p1 is: (3.0, 2.0)
Point p2 is: (6.0, 5.0)
p1 + p2 is: (9.0, 7.0)
p2-p1 is: (3.0, 3.0)
The distance from p1 to p2 is: 4.2X Point-2 (6, 5)

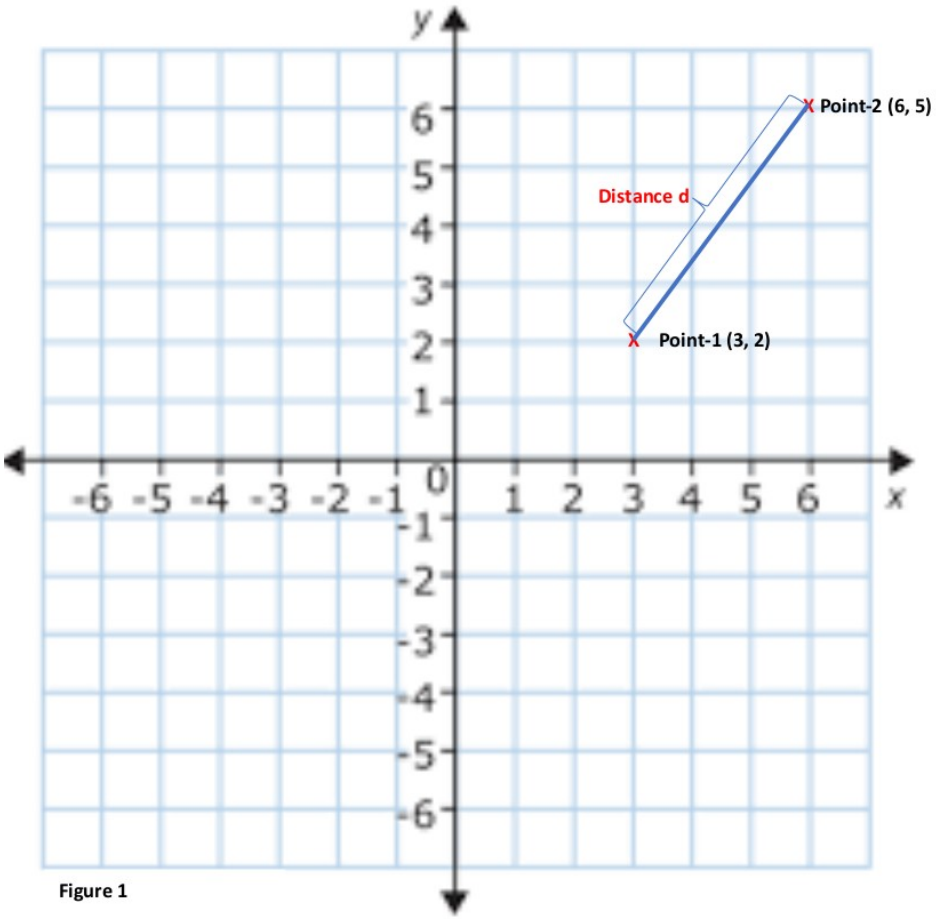

Figure 1

Question 3 : Write a C++ program using operator overloading with the help of a friend function to sum the price of two books. The price of one book is 55 Rupees and 60 paise and the price of the other is 40 Rupees and 70 paise.
The requirement of this program is:
a. To use the operator + and operator = overloading in a proper way.
b. Put a check on the price entry to avoid the negative prices.
c. If the amount in paise after adding the price of both books is greater than 100, then convert it into rupees and add it with the rupees amount.
Desired output of the program:
Invalid Input:
Enter the price of Book1:
Rupees= -50
You have entered Invalid price, please re-enter your price in positive values.
The price of Book1:
Rupees = 55
Paise = 60
The price of Book2:
Rupees = 40
Paise = 70
The total price is = 96 rupees and 30 paise.

Question 4 Write a C++ program to develop a Fraction class and performs arithmetic operations (as shown in the following Fig. 2) with them.

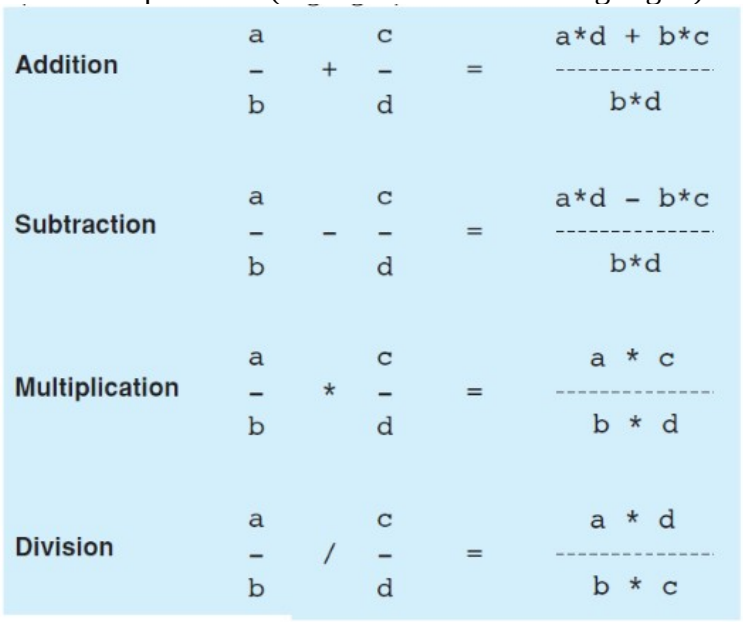

Figure 2

Setup:
a. Define a Fraction class with a numerator and a denominator of type long using a header file called Fraction.h. The constructor has two parameters of type long: the first parameter (numerator) contains the default value 0, and the second parameter (denominator) contains the value 1.
b. Declare operator functions as methods for - (unary), ++ and -- (prefix only), +=, -=, *=, and /=. The operator functions of the binary operators +, -, *, / and the input, output operators <<, >> are to be declared as friend functions of the Fraction class.

c. Implement the constructor for the Fraction class to obtain a positive value for the denominator at all times. If the nominator assumes a value of 0, issue an error message and terminate the program. Then write the operator functions.
d. Then write a main function that calls all the operators in the Fraction class as a test application. Output both the operands and the results.
Desired output: The following output is desired from the same program at the same time. Use the Fraction a (5, 3), b (2) in main for the following;
a = 5/3
b = 2/1
a + b = 11/3
a - b = 1/-3
a * b = 10/3
a / b = 5/6
--a = 2/3
++a = 5/3
And use a += Fraction (1, 2); and a -= Fraction (1, 2); for the following output in main
a+= 1/2; a = 13/6
a-= 1/2; a = -20/-12
-b = 2/-1
And now an input from the user:
Enter a fraction:
Numerator: 5
Denominator! = 0: 0
Error: The denominator is 0
New denominator! = 0: 8
Your input: 5/8

Question 5:
Write a program (implementation part: e.g., Test.cpp is required) to calculate the interest amount, given the amount and interest rate. The amount can be negative, so you need to check on data input. You need to use the operator * overloading and operator = overloading appropriately. You can include any C++ library as per your requirements.
Consider the interface as following (Header file: Test.h):

```
class Test {
        long rupee, paise;
public:
        void print();
        Test();
        Test(long a, long b);
        Test operator* (float rate);
        void operator = (Test a);
};
```

The main function in the driver class (Application.cpp):

```
int main() {
        Test obj1;
        Test obj2(0,0);
        obj1.print();
        obj2 = (obj1 * 0.12);
        cout << "after multiplying by rate 0.12" << endl;
        obj2.print();
        return 0;
}
```

Output:

```
Enter your amount in Rupees: 1000
Enter paise: 0
The amount in Rupees 1000
After multiplying by rate 0.12
The amount of interest in rupees is: 120
```