

**NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS**

CS217 Object Oriented Programming- Fall 2020

ASSIGNMENT- 4

Section (A, B, C, D, and F)

Due Date: Thursday 3rd December 2020 at 11:59 pm on Google Classroom

Instructions:

1. Assignments are to be done individually. You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code you write must be your own and you must understand each part of coding. You are encouraged to seek help from the instructors through email, on google classroom or individually visiting their respective offices.

2. The AIM of this assignment is to practice with operator overloading in C++.

3. No late assignments will be accepted.

4. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code.

6. Plagiarism:

Plagiarism of any kind (copying from others and copying from internet, etc.,) is not allowed. If found plagiarized, you will be awarded zero marks in the assignment. Repeating such an act can lead to strict disciplinary actions and failure in course.

Submission Guidelines:

Dear students we will be using auto-grading tools, so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.

i) For each question in your assignment, make a separate .cpp file and header files .h e.g. for question 1, make q1.cpp, q1.h and so on. Each file that you submit must contains your name, student-id, and assignment # on top of the file in the comments.

ii) Combine all your work in one folder. The folder must contain only .cpp files and header files (.h) if necessary (no binaries, no exe files etc.,).

iii) Run and test your program on a lab machine before submission if possible.

iv) Rename the folder as ROLL-NUM_SECTION (e.g. 19i-0001_B) and compress the folder as a zip file. (e.g. 19i-0001_B.zip).

v) Submit the .zip file on Google Classroom within the deadline.

vi) Submission other than Google Classroom (e.g. email etc.) will not be accepted.

vii) The student is solely responsible to check the final zip files for issues like corrupt file, virus in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason, it will lead to zero marks in the assignment.

Important: please note that your name, roll number and especially section must be correct otherwise it will also lead to zero marks in the assignment.

Questions: Total Marks: 100

Background information

Point addition, subtraction and the computation of distance between two points on a Cartesian Coordinate:

The Fig. 1 shows a cartesian coordinate which has two points **p-1 (3, 2)** and **p-2 (6, 5)**. The sum and difference of the two points will be computed as follows:

- If point-1 is (x1, y1) and point-2 is (x2, y2), then the sum of the point-1 and point-2 will be computed as (x1 + x2, y1 + y2),
- and the difference will be computed as (x2-x1, y2-y1),
- whereas the distance between point-1 and point-2 will be computed according to the Pythagorean formula as follows:

$$p1p2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Note: for more info of cartesian coordinate and Pythagorean formula use this link:

<https://www.mathsisfun.com/algebra/distance-2-points.html>

Question 1 (20 maks): 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. (Marks: 5+5+10)

Question 2 (20 marks): 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. (Marks: 5+5+10)

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).

The desired output in both cases will be as follows:

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: 4.2

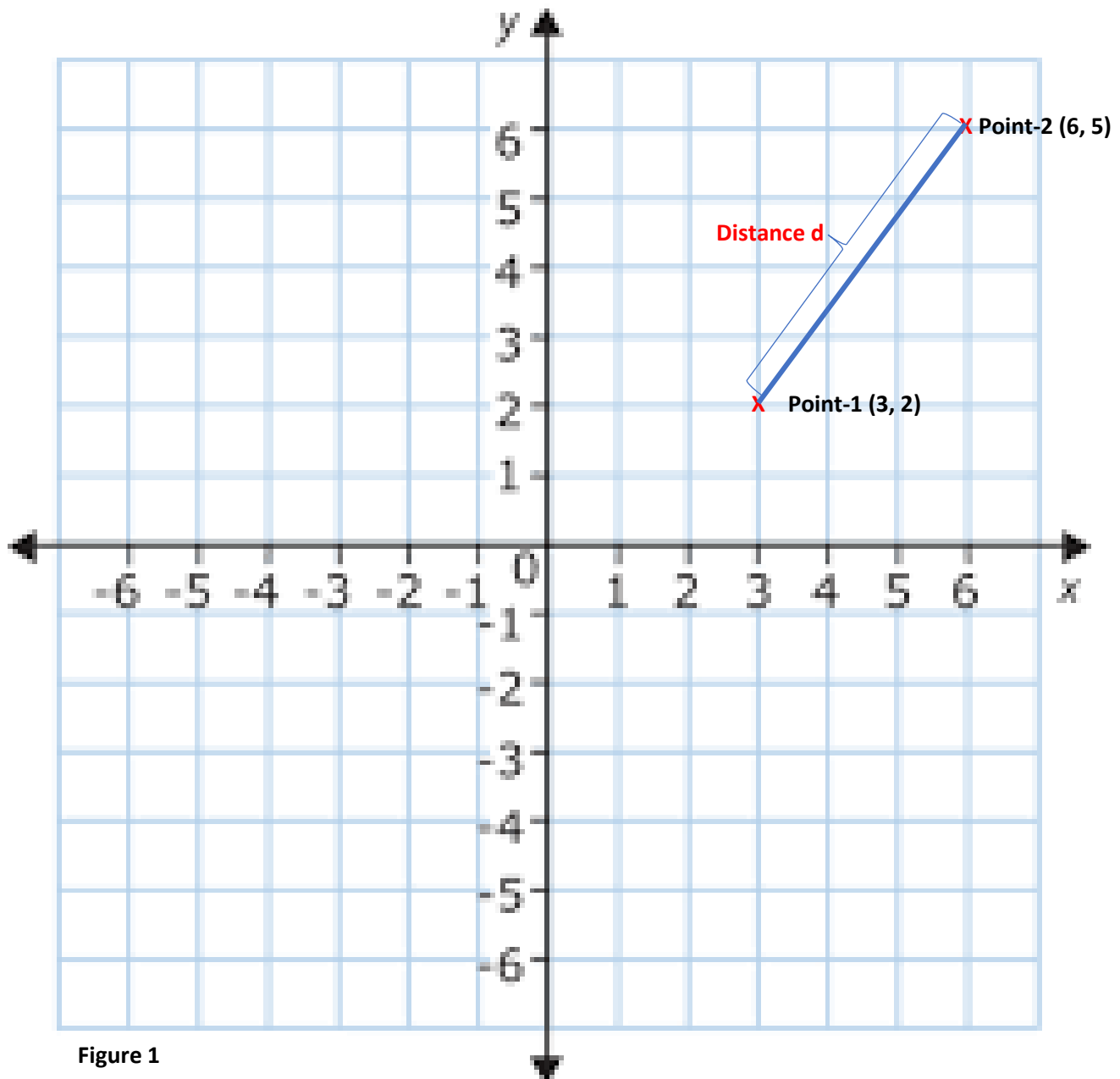


Figure 1

Question 3 (20 marks): 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:

- To use the operator + and operator = overloading in a proper way. (5 Marks)
- Put a check on the price entry to avoid the negative prices. (5 Marks)
- 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 (20 marks): Write a C++ program to develop a Fraction class and performs arithmetic operations (as shown in the following Fig. 2) with them.

Addition	a		c		$\frac{a*d + b*c}{b*d}$
	-	+	-	=	
	b		d		
Subtraction	a		c		$\frac{a*d - b*c}{b*d}$
	-	-	-	=	
	b		d		
Multiplication	a		c		$\frac{a * c}{b * d}$
	-	*	-	=	
	b		d		
Division	a		c		$\frac{a * d}{b * c}$
	-	/	-	=	
	b		d		

Figure 2

Setup:

- 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. (5 marks)
- 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. (5 marks)
- 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. (5 marks)
- 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. (5 marks)

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 (20 marks):

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. (20 Marks)

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