1. Create a class called **Fraction** to perform arithmetic operations with fractional numbers.

Write a program to test this class:

Use integer values to represent the attributes of the class – the numerator and the denominator.

Create **public** methods to perform the following:

a. Input two fractions. The result is normalized.

b. Add two fractions. The result is normalized.

c. Subtract two fractions. The result is normalized.

d. Multiply two fractions. The result is normalized.

e. Divide two fractions. The result is normalized.

f. Display a fraction as follows: (A/B). A is the numerator, B is the denominator

g. Display a fraction on the screen as a decimal number.

2. Using the **Fraction** class of exercise 1. create the class **ArrayOfFractions** which

provides methods to:

a. Input 10 fractions.

b. Find the two fractions which have the greatest sum.

c. Sort the fractions in ascending order.

3. Creating a class called **Point** that has two fields: the abscissa and the ordinate. The **Point** class should include methods that:

a. Constructor to initialize the fields with zero.

b. Input a point.

c. Output a point with form (A;B). A is the abscissa and B is the ordinate.

d. Compute the distance between two points

e. Compute the area of a triangle (defined by 3 points)

4. Use the **Point** class of exercise 3 to create the class **ArrayOfPoints** which provides methods to:

a. Input 10 points

b. Find the largest distance between two points in 10 points.

c. Find a triangle that has largest circumference in 10 points.

5. Create a Candidate class that includes fields:

a. codeID,
b. name,
c. day of birth,
d. test mark 1,
e. test mark 2
f. test mark.

Write methods that compute:

- Input the codeID, the name, the year of birth, the test mark 1, the test mark 2 and the test mark for a student.
- Compute the average mark as follow:

(((the test mark 1 + the test mark 2)/2) + (the test mark *2))/3

Ranks students as follows:

Average mark	Rank
8-10	Good
7 - <8	Fairly good
5 - <7	Average
<5	Fail

6. Use the Candidate class of exercise 4 to create TestCandidate to execute functions as follow:

a. Input 10 candidates

b. How many the students pass?

c. Output the students pass on the screen.