

OOAD
Lab Exercise

Q1. Create a class called **Rational** for performing arithmetic with fractions. Write a program to test your class. Use integer variables to represent the private instance variables of the class—the numerator and the denominator. Provide a constructor that enables an object of this class to be initialized when it's declared. The constructor should store the fraction in reduced form.

The fraction $\frac{2}{4}$ is equivalent to $\frac{1}{2}$ and would be stored in the object as 1 in the numerator and 2 in the denominator. Provide a no-argument constructor with default values in case no initializers are provided. Provide public methods that perform each of the following operations:

- a) Add two Rational numbers: The result of the addition should be stored in reduced form. Implement this as a static method.
- b) Subtract two Rational numbers: The result of the subtraction should be stored in reduced form. Implement this as a static method.
- c) Multiply two Rational numbers: The result of the multiplication should be stored in reduced form. Implement this as a static method.
- d) Divide two Rational numbers: The result of the division should be stored in reduced form. Implement this as a static method.
- e) Return a String representation of a Rational number in the form a/b , where a is the numerator and b is the denominator.
- f) Return a String representation of a Rational number in floating-point format. (Consider providing formatting capabilities that enable the user of the class to specify the number of digits of precision to the right of the decimal point.)

Q2. Create a class **TicTacToe** that will enable you to write a program to play TicTac-Toe. The class contains a private 3-by-3 two-dimensional array. Use an **enum** type to represent the value in each cell of the array. The enum's constants should be named X, O and EMPTY (for a position that does not contain an X or an O). The constructor should initialize the board elements to EMPTY. Allow two human players.

Wherever the first player moves, place an X in the specified square, and place an O wherever the second player moves. Each move must be to an empty square. After each move, determine whether the game has been won and whether it's a draw. If you feel ambitious, modify your program so that the computer makes the moves for one of the players. Also, allow the player to specify whether he or she wants to go first or second. If you feel exceptionally ambitious, develop a program that will play three-dimensional Tic-Tac-Toe on a 4-by-4-by-4 board. (*Optional: Extra credit problem*)