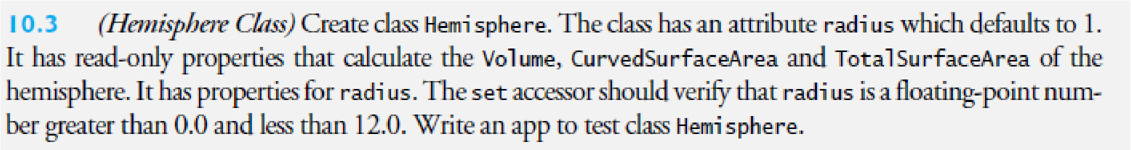
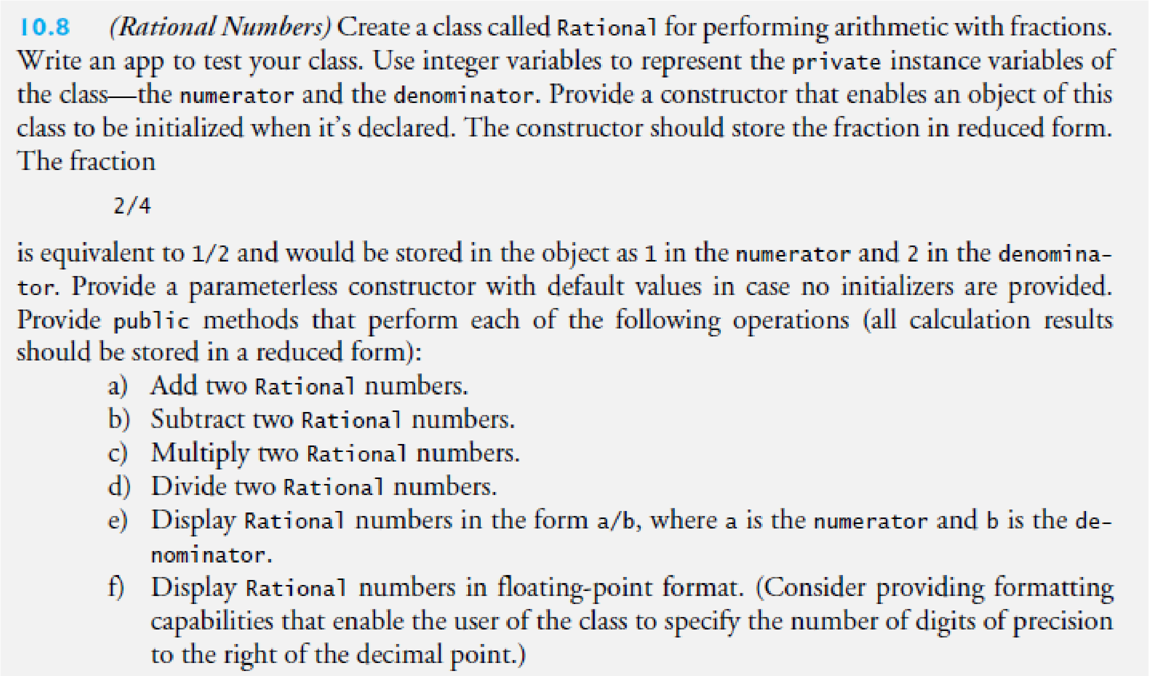
Nesneye yönelik programlama – Ödev 5

Ders kitabının bölüm sonu sorularından olan aşağıdaki soruların her birinin çözümünü ayrı bir Visual Studio konsol uygulaması geliştiriniz.

- Ödevler grup halinde yapılacaktır ve gruptan tek bir kişinin grup numarası isminde bir zip dosyası yüklemesi gerekmektedir (Örnek: 2 numaralı grup 2.zip dosyası yüklemelidir). Birden fazla yükleme olması durumunda tüm grup üyeleri ilgili ödevden 0 notunu alacaktır.
- Ödevlerin kontrolü ödev son tarihinden sonra gelen ilk derste (dersin sonunda) yapılacaktır. Bu nedenle grubun ödevi sunacak/çalıştıracak şekilde sınıfta bulunması gereklidir.



10.3 *(Hemisphere Class)* Create class `Hemisphere`. The class has an attribute `radius` which defaults to 1. It has read-only properties that calculate the `Volume`, `CurvedSurfaceArea` and `TotalSurfaceArea` of the hemisphere. It has properties for `radius`. The `set` accessor should verify that `radius` is a floating-point number greater than 0.0 and less than 12.0. Write an app to test class `Hemisphere`.

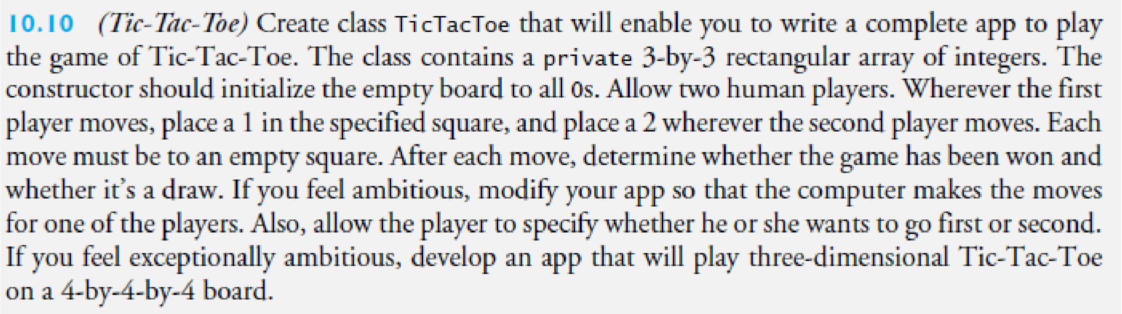


10.8 *(Rational Numbers)* Create a class called `Rational` for performing arithmetic with fractions. Write an app to test your class. Use integer variables to represent the `private` instance variables of the class—the `numerator` and the `denominator`. Provide a constructor that enables an object of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

```
2/4
```

is equivalent to `1/2` and would be stored in the object as `1` in the `numerator` and `2` in the `denominator`. Provide a parameterless constructor with default values in case no initializers are provided. Provide `public` methods that perform each of the following operations (all calculation results should be stored in a reduced form):

a) Add two `Rational` numbers.
b) Subtract two `Rational` numbers.
c) Multiply two `Rational` numbers.
d) Divide two `Rational` numbers.
e) Display `Rational` numbers in the form `a/b`, where `a` is the `numerator` and `b` is the `denominator`.
f) Display `Rational` numbers 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.)



10.10 *(Tic-Tac-Toe)* Create class `TicTacToe` that will enable you to write a complete app to play the game of Tic-Tac-Toe. The class contains a `private` 3-by-3 rectangular array of integers. The constructor should initialize the empty board to all `0`s. Allow two human players. Wherever the first player moves, place a 1 in the specified square, and place a 2 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 app 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 an app that will play three-dimensional Tic-Tac-Toe on a 4-by-4-by-4 board.