

## UTI 526 Object Oriented Programming

### 4<sup>th</sup> Homework

Assignment Date: 01.04.2024

Due Date: 08.04.2024

**From:** Java How to Program, Early Objects (10th Edition), Paul J. Deitel and Harvey Deitel, Pearson, 2015.

**8.6. (*Savings Account Class*)** Create class `SavingsAccount`. Use a static variable `annualInterestRate` to store the annual interest rate for all account holders. Each object of the class contains a private instance variable `savingsBalance` indicating the amount the saver currently has on deposit. Provide method `calculateMonthlyInterest` to calculate the monthly interest by multiplying the `savingsBalance` by `annualInterestRate` divided by 12—this interest should be added to `savingsBalance`. Provide a static method `modifyInterestRate` that sets the `annualInterestRate` to a new value. Write a program to test class `SavingsAccount`. Instantiate two `savingsAccount` objects, `saver1` and `saver2`, with balances of \$2000.00 and \$3000.00, respectively. Set `annualInterestRate` to 4%, then calculate the monthly interest for each of 12 months and print the new balances for both savers. Next, set the `annualInterestRate` to 5%, calculate the next month's interest and print the new balances for both savers. **(40 points)**

**8.15. (*Rational Numbers*)** 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:

- Add two `Rational` numbers: The result of the addition should be stored in reduced form. Implement this as a static method.
- Subtract two `Rational` numbers: The result of the subtraction should be stored in reduced form. Implement this as a static method.
- Multiply two `Rational` numbers: The result of the multiplication should be stored in reduced form. Implement this as a static method.
- Divide two `Rational` numbers: The result of the division should be stored in reduced form. Implement this as a static method.
- Return a `String` representation of a `Rational` number in the form `a/b`, where `a` is the numerator and `b` is the denominator.
- 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.). **(60 points)**

### Important Notes:

1. All source code and related homework reports should be submitted via [Ege Ders](#) platform: 2023 - 2024 Bahar Dönemi → Enstitüler → Fen Bilimleri Enstitüsü → Uluslararası Bilgisayar → Bilgi Teknolojileri ve İnternet Güvenliği → İÖ - Nesne Yönelimli Programlama - 540443 - 2324B → Hafta 7: Classes and Objects: A Deeper Look → Homework 4.

2. Do not forget to include appropriate comments in the source code. Hence the grader can easily understand the program during his/her assessment.
3. Write the programs in a simple and straightforward manner by considering object-oriented analysis and design principles.
4. Each report should include the printout of the related source code, two or more screenshots (depending on the illustration requirements) which exemplify execution of the programs and proper UML diagrams.
5. Homework reports are MANDATORY! Sending only source code without reports including the above mentioned content is subject to getting lower points.
6. IMPORTANT NOTICE: There will be significant point deductions for late, copied or shared submissions.