Lab Task 6 – Classes and Objects – A Deeper Look

1. Create class Rectangle. The class has attributes length and width, each of which defaults to 1. It has read-only properties that calculate the Perimeter and the Area of the rectangle. It has properties for both length and width. The set accessors should verify that length and width are each floating-point numbers greater than 0.0 and less than 20.0. Write an application to test class Rectangle.

2. Create the class SavingsAccount. Use the 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 static method ModifyInterestRate to set the annualInterestRate to a new value. Write an application to test class SavingsAccount. Create two savingsAccount objects, saver1 and saver2, with balances of $2000.00 and $3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and display the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and display the new balances for both savers.

3. Create a class called Complex for performing arithmetic with complex numbers. Complex numbers have the form realPart + imaginaryPart * I where sqrt of i is -1. Write an application to test your class. Use floating-point variables to represent the private data of the class. Provide a constructor that enables an object of this class to be initialized when it's declared. Provide a parameterless constructor with default values in case no initializers are provided. Provide public methods that perform the following operations:

Add two Complex numbers: The real parts are added together and the imaginary parts are added together.

Subtract two Complex numbers: The real part of the right operand is subtracted from the real part of the left operand, and the imaginary part of the right operand is subtracted from the imaginary part of the left operand.

Return a string representation of a Complex number in the form (a, b), where a is the real part and b is the imaginary part.

4. Create a class called Rational for performing arithmetic with fractions. Write an application 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):

- Add two Rational numbers.
- Subtract two Rational numbers.
- Multiply two Rational numbers.
- Divide two Rational numbers.

Display Rational numbers in the form a/b, where a is the numerator and b is the denominator.

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.)

==========SUBMISSION==========

Date: Thursday, April 16, 2020

Time: 9:00 am