CSC 201J-02/Spring 2009 Instructor: Beifang Yi

Assignment 9

(Due date: Wednesday, 4/29/2009 in class)

Your name:	Score:

For each of the following projects, you must first analyze each project, draw a UML Class diagram (you must determine ***how many*** attributes and methods the class has, and ***what*** they are), and then based on the class diagram, implement the project with Java code.

- **Print out your source code and staple it together with the UML class diagrams and this cover paper**.
- Show the result to the instructor **or** compress your PROJECT and send the compressed file to the instructor by email.

1.Rectangle Class: Create a class **Rectangle**. The class has attributes length and width, each of which defaults to 1. It has methods that calculate the perimeter and the area of the rectangle. It has set and get methods for both length and width. The set methods should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0. Complete the following RectangleTest (testing class) program to test class Rectangle (using the sample result output shown below to guide the implementation).

- Draw a UML class diagram for class **Rectangle**:
  - It must contain **2 constructors**: a default with no parameters and a constructor with 2 parameters for width and length.

2.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 $3000.00 and $5000.00, respectively. Set *annualInterestRate* to 5.5%, then calculate the monthly interest and print the new balances for both savers. Then set the *annualInterestRate* to 7.5%, calculate the next month’s interest and print the new balances for both savers. Complete the following SavingAccountTest (testing class) program to test class SavingAccount (using the sample result output shown below to guide the implementation).

- Draw a UML class diagram for class **SavingsAccount**:
    - It must contain **3 constructors**: a default with no parameters, a constructor with **1** parameter for for *savingsBalance*, and a constructor with **2** parameters for *savingsBalance* and *annualInterestRate*.