Assignment 8
(Due date: Friday, 4/24/2009 in class)

Your name:	Score:

For all the following Java projects, you need to:

- **Print out your source code and staple it together with this 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).

```
// RectangleTest.java
// Program tests class Rectangle.
import java.util.Scanner;

public class RectangleTest
{
   public static void main( String args[] )
   {
      Scanner input = new Scanner( System.in );
      Rectangle rectangle = new Rectangle();
      int choice = getMenuChoice();

      while ( choice != 3 )
      {
         switch ( choice )
         {
            case 1:
               System.out.print( "Enter length: " );
               rectangle.setLength( input.nextDouble() );
               break;

            case 2:
```

```
                System.out.print ( "Enter width: " );
                rectangle.setWidth( input.nextDouble() );
                break;
            } // end switch

            System.out.println ( rectangle.toString() );

            choice = getMenuChoice();
         } // end while
      } // end main

      // prints a menu and returns a value coressponding to the menu choice
      private static int getMenuChoice()
      {
         Scanner input = new Scanner( System.in );
         … …
         return input.nextInt();
      } // end method getMenuChoice
   } // end class RectangleTest
```

```
1. Set Length
2. Set Width
3. Exit
Choice: 1
Enter length: 10
Length: 10.000000
Width: 1.000000
Perimeter: 22.000000
Area: 10.000000
1. Set Length
2. Set Width
3. Exit
Choice: 2
Enter width: 15
Length: 10.000000
Width: 15.000000
Perimeter: 50.000000
Area: 150.000000
1. Set Length
2. Set Width
3. Exit
Choice: 1
Enter length: 99
Length: 1.000000
Width: 15.000000
Perimeter: 32.000000
Area: 15.000000
1. Set Length
2. Set Width
3. Exit
Choice: 3
```

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 $2000.00 and $3000.00, respectively. Set *annualInterestRate* to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the *annualInterestRate* to 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).

```
Monthly balances for one year at .04
Balances:
             Saver 1   Saver 2
Base        $2000.00  $3000.00
Month 1:    $2006.67  $3010.00
Month 2:    $2013.36  $3020.03
Month 3:    $2020.07  $3030.10
Month 4:    $2026.80  $3040.20
Month 5:    $2033.56  $3050.33
Month 6:    $2040.33  $3060.50
Month 7:    $2047.14  $3070.70
Month 8:    $2053.96  $3080.94
Month 9:    $2060.81  $3091.21
Month 10:   $2067.68  $3101.51
Month 11:   $2074.57  $3111.85
Month 12:   $2081.48  $3122.22

After setting interest rate to .05
Balances:
Saver 1      Saver 2
$2090.16    $3135.23
```

```
// SavingAccountTest.java
// Program that tests SavingAccount class

public class SavingAccountTest
{
   public static void main( String args[] )
   {
      SavingAccount saver1 = new SavingAccount( 2000 );
      SavingAccount saver2 = new SavingAccount( 3000 );
      SavingAccount.modifyInterestRate( 0.04 );

      System.out.println( "Monthly balances for one year at .04" );
      System.out.println( "Balances:" );

      System.out.printf( "… … … …", "Saver 1", "Saver 2" );
      System.out.printf("… … … …", "Base",
         saver1.toString(), saver2.toString() );

      for ( int month = 1; month <= 12; month++ )
      {
         String monthLabel = String.format( "Month %d:", month );
         saver1.calculateMonthlyInterest();
         saver2.calculateMonthlyInterest();

         System.out.printf("… … … …", monthLabel,
            saver1.toString(), saver2.toString() );
      } // end for

      SavingAccount.modifyInterestRate( .05 );
      saver1.calculateMonthlyInterest();
      saver2.calculateMonthlyInterest();

      System.out.println( "\nAfter setting interest rate to .05" );
      System.out.println( "Balances:" );
      System.out.printf("… … … …", "Saver 1", "Saver 2" );
      System.out.printf("… … … …",
         saver1.toString(),  saver2.toString() );
   } // end main
} // end class SavingAccountTest
```