Topic: Objects and Classes: Objects vs classes.

Review

Problem 1. **10 point**

(Rectangle Class) Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide methods that calculate the rectangle's perimeter and area. 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. Write a program to test class Rectangle.

Problem 2. **10 point**

(Invoice Class) Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables—a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write a test program that demonstrates class Invoice's capabilities.

Problem 3. **10 point**

(Stopwatch) Design a class named StopWatch. The class contains:

- Private data fields startTime and endTime with getter methods.
- A no-arg constructor that initializes startTime with the current time.
- A method named start() that resets the startTime to the current time.
- A method named stop() that sets the endTime to the current time.
- A method named getElapsedTime() that returns the elapsed time for the stopwatch in milliseconds.

Write a test program that measures the execution time of sorting 100,000 numbers using selection sort.

```
    public class SelectionSort {
    public static void sort(int[] arr) {
        int n = arr.length;
        for (int i = 0; i < n - 1; i++) {
            int minIndex = i;
            for (int j = i + 1; j < n; j++) {
                if (arr[j] < arr[minIndex]) {
                    minIndex = j;
                }
            }
            int temp = arr[minIndex];
            arr[minIndex] = arr[i];
            arr[i] = temp;
        }
    }
}
```

Problem 4. **10 point**

(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 *savings-Balance*.
- 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.