/89 Name: _____ Period: _____

APCS Problem Set 9: A Second Look at OO Programming

2.2 Enhancing `Fraction`

Download `Fraction.java` from http://feromax.com/apcs/problemsets/PS09/downloads/ and import into a new project called `PS9Fraction`. Per Litvin Ch. 10, #6, add the specified methods. Once done, convince yourself that your class works properly by testing the following fraction operations.

1. Subtracting two fractions: $\frac{5}{11} - \frac{27}{11} \stackrel{?}{=} -2$
2. Dividing two fractions: $\frac{2}{3} \div \frac{3}{2} \stackrel{?}{=} \frac{4}{9}$
3. Dividing a fraction by another whose numerator is zero: $\frac{2}{3} \div \frac{0}{1} \leftarrow$ `divide()` method generates an exception (i.e., the `Fraction()` constructor isn't the method throwing the exception)

In order to get your code signed off, you will need to run a prepared tester class that will choose random fractions and utilize the methods you wrote. `FractionTester.class` is available from http://feromax.com/apcs/problemsets/PS09/downloads/FractionTester/. Save it in your Eclipse project's `bin` directory (so it's in the same directory as `Fraction.class`), start a terminal shell, and run the tester:

```
java FractionTester
```

Teacher's Initials: _____ (12pts)

10 `SnackBar`, Part I

10.2 Implementing `Vendor.java`

10.2.5 Testing the methods

Make sure your tester outputs what it's doing at every step before getting a sign-off.

Teacher's Initials: _____ (20pts)

10.3 Putting It All Together

Download `coin.gif`, `SnackBar.java`, and `VendingMachine.java` from http://feromax.com/apcs/problemsets/PS09/downloads/SnackBar/, importing each into your project. Run class `SnackBar` and debug your code as necessary to get the program to work like the provided JAR version. Afterward, demonstrate your working `SnackBar` for a sign-off.

Teacher's Initials: _____ (10pts)

11 `Time` Class Exercise

Per Litvin Ch. 10, #13, write and test the `Time` class. The source for the tester class referred to in part (f), `TestTime.java`, may be downloaded from http://feromax.com/apcs/problemsets/PS09/downloads/. You might use the provided `TestTime` class to check your `elapsedTime()` method using these specific values:

1. t_1 = `11:30AM`, t_2 = `1:30PM` → `t2.elapsedSince(t1)` should return `120` minutes
2. t_1 = `12:02AM`, t_2 = `12:01AM` → `t2.elapsedSince(t1)` should return `1439` minutes
3. t_1 = `26:03` → should generate an `IllegalArgumentException`

Once you've done some preliminary testing using Litvin's interactive `TestTime` class, you will need to run `BetterTestTime`, available from http://feromax.com/apcs/problemsets/PS09/downloads/TimeAutoTester/. Just as with the `FractionTester` class, save `BetterTimeTester.class` in your Eclipse project's `bin` directory and run it from the terminal shell.

If you cannot get all tests to pass, you'll be given partial credit according to the tester's final score.

Teacher's Initials: _____ _____/32pts

14 `SnackBar`, Part II

When you read the first part of Litvin §10.12, an elegant solution is outlined in which a new `BookKeeper` class is described. For those who finish the solution that uses `static`, which is described in the second half of §10.12, you are encouraged to implement the more elegant solution.

Modify your `SnackBar` as described in the second half of Litvin §10.12 and demonstrate the new functionality for a sign-off.

Teacher's Initials: _____ (15pts)

16 Bonus Round: `Coins` Class

For bonus credit, complete Litvin Ch. 10, #16.

Teacher's Initials: _____ (+5pts)