COS 126 **General Computer Science** **Spring 2011**

Programming Exam 2

This test has 1 question. You have 50 minutes. The exam is open book, open note, and open web. You may use code from your programming assignments or the *Introduction to Programming in Java* booksite. No communication with any non-staff members is permitted. Submit your solution via Dropbox. **Write out and sign the Honor Code pledge before turning in the test.**

"I pledge my honor that I have not violated the Honor Code during this examination."

Name:

Signature

NetID:

Total	

P01	TTh 1:30	Keith
P01A	TTh 1:30	Doug
P01B	TTh 1:30	Victor
P01C	TTh 1:30	Richard
P01D	TTh 1:30	Gordon
P01E	TTh 1:30	Arman
P02	TTh 2:30	Doug
P03	TTh 3:30	Gordon
P03A	TTh 3:30	Keith
P04	TTh 7:30	Nick
P05	WF 10	Dmitry
P06	WF 1:30	Victor
P06A	WF 1:30	Chris
P06B	WF 1:30	Donna
P07	WF 12:30	Donna

Do not remove this exam from the room.

Problem. Write a data type `LR.java` that models a linear relationship between a response variable y and a predictor variable x using *simple linear regression.* Suppose there are n observation pairs (x_i, y_i) for $i = 1$ to n. The goal is to find the coefficients a and b of the straight line

$$y \;=\; ax \;+\; b$$

that "best" fits the observations. We give the formulas for the *least squares* solution below.

- The *means* of the x_i and y_i values are defined as:

$$\overline{x} \;=\; \frac{x_1 + x_2 + \ldots + x_n}{n}, \qquad \overline{y} \;=\; \frac{y_1 + y_2 + \ldots + y_n}{n}$$

- The intermediate terms S_{xx} and S_{xy} are defined as:

$$S_{xx} \;=\; (x_1 - \overline{x})(x_1 - \overline{x}) + (x_2 - \overline{x})(x_2 - \overline{x}) + \ldots + (x_n - \overline{x})(x_n - \overline{x})$$

$$S_{xy} \;=\; (x_1 - \overline{x})(y_1 - \overline{y}) + (x_2 - \overline{x})(y_2 - \overline{y}) + \ldots + (x_n - \overline{x})(y_n - \overline{y})$$

- The *slope* a and *y-intercept* b of the best-fit line are:

$$a \;=\; S_{xy} \,/\, S_{xx}\;, \qquad b \;=\; \overline{y} - a\,\overline{x}$$

Example. For example, suppose that $n = 4$ and the observation pairs are:

i	x_i	y_i
1	20	91
2	40	83
3	60	68
4	80	50

Then, the best-fit line is $y = -0.69x + 107.50$. Below are the step-by-step calculations.

$$\overline{x} \;=\; \frac{20 + 40 + 60 + 80}{4} \;=\; 50, \qquad \overline{y} \;=\; \frac{91 + 83 + 68 + 50}{4} \;=\; 73$$

$$S_{xx} \;=\; (20-50)(20-50) + (40-50)(40-50) + (60-50)(60-50) + (80-50)(80-50) \;=\; 2000$$

$$S_{xy} \;=\; (20-50)(91-73) + (40-50)(83-73) + (60-50)(68-73) + (80-50)(50-73) \;=\; -1380$$

$$a \;=\; -1380/2000 \;=\; -0.69, \qquad b \;=\; 73 - (-0.69)(50) \;=\; 107.50$$

Predicting. Given a predictor variable x_0, the model predicts that the corresponding response variable is $\hat{y}_0 = ax_0 + b$. For example, if $x_0 = 50$, we predict $\hat{y}_0 = -0.69(50) + 107.5 = 73.0$. The following table shows the predictor variables, the observed responses, and the responses predicted by the model.

i	x_i	y_i	$\hat{y}_i$
1	20	91	93.70
2	40	83	79.90
3	60	68	66.10
4	80	50	52.30

API specification. Organize your program `LR.java` as a data type with the following API:

public class LR		
`public`	`LR(double[] x, double[] y)`	*linear regression with* (x_i, y_i) *throw an exception if lengths are not equal*
`public`	`double meanx()`	*mean of the* x_i *values*
`public`	`double meany()`	*mean of the* y_i *values*
`public`	`double slope()`	*slope* a *of best-fit line*
`public`	`double intercept()`	*y-intercept* b *of best-fit line*
`public`	`double predict(double x0)`	*estimate* $\hat{y}_0 = ax_0 + b$
`public`	`static void main(String[] args)`	*read data from standard input, prints results to standard output, as described below*

Input and output specification. The `main()` function should read in a sequence of observation pairs from standard input, compute the best-fit line, and print out the observation pairs and the predicted values. Your `main()` must read input and write output as directed below:

- *Standard input.* An integer N followed by N observation pairs of (x_i, y_i) real values.
- *Standard output.* The best-fit line, followed by N lines of output, where each line contains x_i, y_i, and $\hat{y}_i$. Each number should be formatted with two digits after the decimal place.

Assume that $N \geq 2$ and that at least two of the x_i values are distinct to ensure that $S_{xx} \neq 0$.

```
% more lr4.txt              % java LR < lr4.txt
4                           y = -0.69 x + 107.50
 20.0  91.0                 20.00  91.00  93.70
 40.0  83.0                 40.00  83.00  79.90
 60.0  68.0                 60.00  68.00  66.10
 80.0  50.0                 80.00  50.00  52.30
```

For convenience, the following test input files are available:

```
http://introcs.cs.princeton.edu/data/lr4.txt
http://introcs.cs.princeton.edu/data/lr10.txt
http://introcs.cs.princeton.edu/data/lr1000.txt
```

```
% more lr10.txt          % java LR < lr10.txt
10                       y = 2.00 x + 34.57
 26.32  87.70            26.32  87.70  87.34
 14.17  62.71            14.17  62.71  62.98
 18.37  73.12            18.37  73.12  71.40
 29.76  94.07            29.76  94.07  94.24
 15.01  64.99            15.01  64.99  64.67
 25.98  85.01            25.98  85.01  86.66
 13.04  60.47            13.04  60.47  60.72
 14.25  62.04            14.25  62.04  63.14
 14.31  63.57            14.31  63.57  63.26
 27.98  91.39            27.98  91.39  90.67

% more lr1000.txt        % java LR < lr1000.txt
1000                     y = -0.50 x + 55.11
 58.68  24.46            58.68  24.46  25.65
 49.80  28.88            49.80  28.88  30.10
 39.52  36.08            39.52  36.08  35.27
 41.27  34.91            41.27  34.91  34.39
 30.22  38.91            30.22  38.91  39.93
 54.52  28.96            54.52  28.96  27.73
 35.03  38.62            35.03  38.62  37.52
...                      ...
```

Submission. Submit `LR.java` via Dropbox at

`https://dropbox.cs.princeton.edu/COS126_S2011/Exam2`

Be sure to click the *Check All Submitted Files* button to verify your submission.

Grading. *Your program will be graded on correctness and clarity (including comments). You will receive partial credit for correctly implementing the following components:*

- *The* `meanx()` *and* `meany()` *methods.*
- *The* `slope()` *and* `intercept()` *methods.*
- *The* `predict()` *methods.*
- *Reading the input data, storing it in two parallel arrays, and printing it back out.*

You will receive a substantial penalty if your program does not compile or if you do not follow the prescribed API or input/output specifications.