# Hands-on Experiment # 6 : Worksheet

Section\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

No more than 3 students per one submission of this worksheet.

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Part A: Loop Writing Practice

In *MathPowLoop.java*, write Java statements using “loops” to calculate result2 so that its value is similar to result1 (which is calculated from *Math.pow()* ) for every double a and int b.

No methods in the *Math* class is allowed.

List your code here.

import java.util.Scanner;

import java.io.\*;

import java.lang.Math;

public class PartA {

public static void main(String[] *args*) {

double a = 2.0;

int b = 8;

double result1 = Math.pow(a, b);

double result2 = 1;

for (int i = 0; i < Math.abs(b); i++)

result2 \*= a;

if (b < 0)

result2 = 1 / result2;

System.out.println(result1);

System.out.println(result2);

}

}

Test your code with the following test data set.

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | Math.pow(a,b) | Your code |
| 2.0 | 8 | 256.0 | 256.0 |
| 2.5 | 3 | 15.625 | 15.625 |
| -2.0 | 8 | 256.0 | 256.0 |
| 1.0 | 1 | 1.0 | 1.0 |
| 1.0 | 0 | 1.0 | 1.0 |
| 2.0 | 30 | 1.073741824E9 | 1.073741824E9 |
| -2.0 | 30 | 1.073741824E9 | 1.073741824E9 |
| 2.0 | -1 | 0.5 | 0.5 |
| 2.0 | -4 | 0.0625 | 0.0625 |

## Part B: Text File Processing

The file *score.csv* contains scores from the midterm examination of a programming course, which has 5 questions (Q1-Q5). The file is in the “Comma-separated Value” format (<http://en.wikipedia.org/wiki/Comma-separated_values>) with the first line being the header labels describing the order of data on the other lines.

* Read <http://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html> to learn how to read a text file using an instance of the Scanner class.
* Open the file in a spreadsheet application (such as MS Excel). If you do not have any spreadsheet application on your machine, try using Google Spreadsheet.
  + Use the application to find the average score, the maximum score, and the minimum score of each question (Q1-Q5).
  + Find the average of the total score and its corresponding standard deviation.
* Fill the results in the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| From Spreadsheet | Average | Standard Deviation | Max | Min |
| Q1 |  |  |  |  |
| Q2 |  |  |  |  |
| Q3 |  |  |  |  |
| Q4 |  |  |  |  |
| Q5 |  |  |  |  |
| Total |  |  |  |  |

* Write a Java program to:
  + Compute the average score, the maximum score, and the minimum score of each question (Q1-Q5).
  + Compute the average of the total score and its corresponding standard deviation.
* Fill the results in the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| From Your Java App | Average | Standard Deviation | Max | Min |
| Q1 |  |  |  |  |
| Q2 |  |  |  |  |
| Q3 |  |  |  |  |
| Q4 |  |  |  |  |
| Q5 |  |  |  |  |
| Total |  |  |  |  |

List your code here.

import java.util.Scanner;

import java.io.\*;

import java.util.Arrays;

public class PartB {

public static void main(String[] *args*) throws IOException {

Scanner in = new Scanner(new File("score.csv"));

in.nextLine();

in.useDelimiter("\\r\\n|,");

int q[][] = new int[1000][5];

double sumpowdiff[] = new double[5];

int sum[] = new int[5];

int max[] = new int[5];

int min[] = new int[5];

int num = 0;

double count = 0;

double avg[] = new double[5];

double sd[] = new double[5];

double sumavgpowdiff = 0;

double sumavg = 0;

double avgq, sdq;

double totalsumpowdiff = 0;

Arrays.fill(max, -99999);

Arrays.fill(min, 99999);

while (in.hasNext()) {

String id = in.next();

for (int i = 0; i < 5; i++) {

q[num][i] = in.nextInt();

sum[i] += q[num][i];

if (q[num][i] > max[i])

max[i] = q[num][i];

if (q[num][i] < min[i])

min[i] = q[num][i];

}

count++;

num++;

}

for (int i = 0; i < 5; i++) {

avg[i] = sum[i] / count;

System.out.println("Avg for Q" + (i + 1) + ": " + avg[i]);

System.out.println("Min = " + min[i] + " Max = " + max[i]);

for (int j = 0; j < 1000; j++) {

sumpowdiff[i] += Math.pow(q[j][i] - avg[i], 2);

totalsumpowdiff += Math.pow(q[j][i] - avg[i], 2);

}

sumavg += avg[i];

sd[i] = Math.sqrt(sumpowdiff[i] / 1000);

System.out.println("SD For Q" + (i + 1) + ": " + sd[i]);

}

avgq = sumavg / 5;

System.out.println("Avg for all Q : " + avgq);

for (int i = 0; i < 5; i++)

for (int j = 0; j < 1000; j++)

sumavgpowdiff += Math.pow(q[j][i] - avgq, 2);

sdq = Math.sqrt(sumavgpowdiff / 5000);

System.out.println("SD for all Q : " + sdq);

}

}

Submit this worksheet (by only one member of the group) via <http://www.myCourseVille.com> (Assignments > Hands-on Experiment # 6) **within the day after your lecture**.