Homework Ten

A201/A597/I210—Spring Semester 2005

Due the week of April 28-29

Abstract

Read and solve the problems below. Turn them in to your lab instructor in person during the last lab or office hours.

The Computer Science Department[1] and the School of Informatics[2] clearly specify the rules of academic honesty and academic integrity, so please read the documents and make sure you comply.

Posting solutions or major hints on the bulletin board is not allowed.

1 The Problems

1. a) Write a Java program to input the following integer numbers into an array named `grades`: 89, 95, 72, 83, 99, 54, 86, 75, 92, 73, 79, 75, 82, 73. As each number is input, add the number to a total. After all numbers are input and the total is obtained, calculate the average of the numbers and use the average to determine the deviation of each value from the average. Store each deviation in an array named `deviation`. Each deviation is obtained as the element values less the average of all the data. Have your program display each deviation alongside its corresponding element from the `grades` array.

   b) Calculate the variance of the data used in Exercise 1-a). The variance is obtained by squaring each individual and dividing the sum of the squared deviations by the number of deviations.

2. a) Store the following data into a file:

   ```
   5 96 87 78 93 21 4 92 82 85 87 6 72 69 85 75 81 73
   ```

   b) Write a Java program to calculate and display the average of each group of numbers in the file created in Exercise 2-a). The data are arranged in the file so that each group of numbers is preceded by the number of data items in the group. Thus, the first number in the file, 5, indicates that the next five numbers should be grouped together. The number 4 indicates that the following four numbers are a group, and 6 indicates that the last six numbers are a group. (Hint: Use a nested loop.)

[1] `http://www.cs.indiana.edu/Academics/integrity.html`
[2] `http://www.informatics.indiana.edu/courses/honesty.asp`

3. Write an application that converts miles to kilometers. (One mile equals 1.60935 kilometers.) Read the miles value from the user as a command-line argument (expressed as a floating point value,) and print the converted value back to the screen.

4. Design and implement a Java program that that determines and prints the number of odd, even, and zero digits in an integer value read from the keyboard (or as a command-line argument).

5. Design and implement a class called `Coin` that represents a coin that can be flipped, showing either heads or tails. Create a driver class, called `CoinFlip`, whose main method flips a coin 100 times to determine how many times each side comes up, and test your class:

```
Coin coin = new Coin();
for (int i = 0; i < 100; i++)
  coin.flip();
coin.report();
```

6. Create three programs that print the three patterns below (as scalable patterns, with the size specified on the command line):

```
* * * * * * * * * * *       * * * * *     * * * * * * * * * * *
* * * * * * * * * *         * * * * *     * * * *     * * * *
* * * * * * * * *           * * * * *     * * *         * * *
* * * * * * *               * * * * *     * *             * *
* * * * * *                 * * * * *     *                 *
* * * * *       * * * * *               *                 *
* * * *         * * * * *               * *             * *
* * *           * * * * *               * * *         * * *
* *             * * * * *               * * * *     * * * *
*               * * * * *               * * * * * * * * * * *
```

7. Design and implement a Java program that reads an integer value representing a year from the user. The purpose of the year is to determine if the year is a leap year (and therefore has 29 days in February) in the Gregorian calendar. A year is a leap year if it is divisible by 4, unless it is also divisible by 100 but not by 400. For example, the year 2003 is not a leap year, but 2004 is. The year 1900 is not a leap year, because it is divisible by 100, but the year 2000 is a leap year because even though it is divisible by 100, it is also divisible by 400. Produce an error message for any input value less than 1582 (the year the Gregorian calendar was adopted).

8. Design and implement an application that plays the Hi-Lo guessing game with numbers. The program should pick a random number between 1 and 100 (inclusive), then repeatedly prompt the user to guess the number. On each guess, report to the user that he or she is correct or that the guess is high or low. Continue accepting guesses until the user guesses correctly or chooses to quit (the user types `done`). Count the number of guesses and report the value when the user guesses correctly. At the end of each game (by quitting or a correct guess), prompt to determine whether the user wants to play again. Continue playing games until the user chooses to stop.

9. Design and implement an application that creates a histogram that allows you to visually inspect the frequency distribution of a set of values. The program should read in an arbitrary number of integers that are in the range 1 to 100 inclusive; then produce a chart similar to the one below that indicates how many input values fell in the range 1 to 10, 11 to 20, and so on. Print one asterisk for each value entered and the total number in parentheses at the end.

```
1  - 10    *****(5)
11 - 20    **(2)
21 - 30    ******************(19)
31 - 40
41 - 50    ***(3)
51 - 60    ********(8)
61 - 70    **(2)
71 - 80    *****(5)
81 - 90    *******(7)
91 - 100   *********(9)
```