Programming Exercises for Chapter 2

THIS PAGE CONTAINS several exercises for Chapter 2 in Introduction to Programming Using Java. For each exercise, a link to a possible solution is provided. Each solution includes a discussion of how a programmer might approach the problem and interesting points raised by the problem or its solution, as well as complete source code of the solution.

Exercise 2.1:

Write a program that will print your initials to standard output in letters that are nine lines tall. Each big letter should be made up of a bunch of *'s. For example, if your initials were "DJE", then the output would look something like:

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

See the Solution

Exercise 2.2:

Write a program that simulates rolling a pair of dice. You can simulate rolling one die by choosing one of the integers 1, 2, 3, 4, 5, or 6 at random. The number you pick represents the number on the die after it is rolled. As pointed out in Section 2.5, the expression

```
(int)(Math.random()*6) + 1
```

does the computation to select a random integer between 1 and 6. You can assign this value to a variable to represent one of the dice that are being rolled. Do this twice and add the results together to get the total roll. Your program should report the number showing on each die as well as the total roll. For example:

```
The first die comes up 3
The second die comes up 5
Your total roll is 8
```

See the Solution

Exercise 2.3:

Write a program that asks the user's name, and then greets the user by name. Before outputting the user's name, convert it to upper case letters. For example, if the user's name is Fred, then the program

should respond "Hello, FRED, nice to meet you!".

Exercise 2.4:

Write a program that helps the user count his change. The program should ask how many quarters the user has, then how many dimes, then how many nickels, then how many pennies. Then the program should tell the user how much money he has, expressed in dollars.

Exercise 2.5:

If you have N eggs, then you have N/12 dozen eggs, with N%12 eggs left over. (This is essentially the definition of the / and % operators for integers.) Write a program that asks the user how many eggs she has and then tells the user how many dozen eggs she has and how many extra eggs are left over.

A gross of eggs is equal to 144 eggs. Extend your program so that it will tell the user how many gross, how many dozen, and how many left over eggs she has. For example, if the user says that she has 1342 eggs, then your program would respond with

```
Your number of eggs is 9 gross, 3 dozen, and 10
```

since 1342 is equal to 9*144 + 3*12 + 10.

Exercise 2.6:

This exercise asks you to write a program that tests some of the built-in subroutines for working with Strings. The program should ask the user to enter their first name and their last name, separated by a space. Read the user's response using `TextIO.getln()`. Break the input string up into two strings, one containing the first name and one containing the last name. You can do that by using the `indexOf()` subroutine to find the position of the space, and then using `substring()` to extract each of the two names. Also output the number of characters in each name, and output the user's initials. (The initials are the first letter of the first name together with the first letter of the last name.) A sample run of the program should look something like this:

```
Please enter your first name and last name, separated by a space.
? Mary Smith
Your first name is Mary, which has 4 characters
Your last name is Smith, which has 5 characters
Your initials are MS
```

Exercise 2.7:

Suppose that a file named "testdata.txt" contains the following information: The first line of the file is the name of a student. Each of the next three lines contains an integer. The integers are the student's

scores on three exams. Write a program that will read the information in the file and display (on standard output) a message that contains the name of the student and the student's average grade on the three exams. The average is obtained by adding up the individual exam grades and then dividing by the number of exams.