

**SYS5110 – Foundations on Modelling and Simulation**  
**Java Exercises, Liang Chapters 1, 2 and 3**  
**Fall 2015**

**1.7** (Approximate  $\pi$ )  $\pi$  can be computed using the following formula:

$$\pi = 4 \times \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots \right)$$

Write a program that displays the result of  $4 \times \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} \right)$  and  $4 \times \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} \right)$ . Use **1.0** instead of **1** in your program.

**\*1.13** (Algebra: solve  $2 \times 2$  linear equations) You can use Cramer's rule to solve the following  $2 \times 2$  system of linear equation:

$$\begin{array}{rcl} ax + by = e & x = \frac{ed - bf}{ad - bc} & y = \frac{af - ec}{ad - bc} \\ cx + dy = f \end{array}$$

Write a program that solves the following equation and displays the value for  $x$  and  $y$ :

$$3.4x + 50.2y = 44.5$$

$$2.1x + .55y = 5.9$$

1770848 2014/08/24 1:55:10 155 18

**\*\*2.13** (Financial application: compound value) Suppose you save **\$100** each month into a savings account with the annual interest rate 5%. Thus, the monthly interest rate is  $0.05/12 = 0.00417$ . After the first month, the value in the account becomes

$$100 * (1 + 0.00417) = 100.417$$

After the second month, the value in the account becomes

$$(100 + 100.417) * (1 + 0.00417) = 201.252$$

After the third month, the value in the account becomes

$$(100 + 201.252) * (1 + 0.00417) = 302.507$$

and so on.

Write a program that prompts the user to enter a monthly saving amount and displays the account value after the sixth month. (In Exercise 5.30, you will use a loop to simplify the code and display the account value for any month.)



Enter the monthly saving amount: **100**   
 After the sixth month, the account value is \$608.81

- \*2.23** (*Cost of driving*) Write a program that prompts the user to enter the distance to drive, the fuel efficiency of the car in miles per gallon, and the price per gallon, and displays the cost of the trip. Here is a sample run:



```
Enter the driving distance: 900.5
Enter miles per gallon: 25.5
Enter price per gallon: 3.55
The cost of driving is $125.36
```

think before coding



#### Pedagogical Note

For each exercise, carefully analyze the problem requirements and design strategies for solving the problem before coding.



#### Debugging Tip

Before you ask for help, read and explain the program to yourself, and trace it using several representative inputs by hand or using an IDE debugger. You learn how to program by debugging your own mistakes.

learn from mistakes

- \*\*3.15** (*Game: lottery*) Revise Listing 3.8, `Lottery.java`, to generate a lottery of a three-digit number. The program prompts the user to enter a three-digit number and determines whether the user wins according to the following rules:

1. If the user input matches the lottery number in the exact order, the award is \$10,000.
2. If all digits in the user input match all digits in the lottery number, the award is \$3,000.
3. If one digit in the user input matches a digit in the lottery number, the award is \$1,000.

### Comprehensive

**\*\*3.21** (*Science: day of the week*) Zeller's congruence is an algorithm developed by Christian Zeller to calculate the day of the week. The formula is

$$h = \left( q + \frac{26(m+1)}{10} + k + \frac{k}{4} + \frac{j}{4} + 5j \right) \% 7$$

where

- **h** is the day of the week (0: Saturday, 1: Sunday, 2: Monday, 3: Tuesday, 4: Wednesday, 5: Thursday, 6: Friday).
- **q** is the day of the month.
- **m** is the month (3: March, 4: April, ..., 12: December). January and February are counted as months 13 and 14 of the previous year.
- **j** is the century (i.e.,  $\frac{year}{100}$ ).
- **k** is the year of the century (i.e.,  $year \% 100$ ).

Note that the division in the formula performs an integer division. Write a program that prompts the user to enter a year, month, and day of the month, and displays the name of the day of the week. Here are some sample runs:



```
Enter year: (e.g., 2012): 2015 ↵ Enter
Enter month: 1-12: 1 ↵ Enter
Enter the day of the month: 1-31: 25 ↵ Enter
Day of the week is Sunday
```



```
Enter year: (e.g., 2012): 2012 ↵ Enter
Enter month: 1-12: 5 ↵ Enter
Enter the day of the month: 1-31: 12 ↵ Enter
Day of the week is Saturday
```