

# Calculating Sales Commission

Hal's Home Computer Emporium is a retail seller of home computers. Hal's sales staff works strictly on commission. At the end of the month, each salesperson's commission is calculated according to Table CS1-1.

**Table CS1-1** Sales commission rates

Sales This Month	Commission Rate
less than \$10,000	5%
\$10,000–14,999	10%
\$15,000–17,999	12%
\$18,000–21,999	15%
\$22,000 or more	16%

For example, a salesperson with \$16,000 in monthly sales will earn a 12% commission (\$1,920). Another salesperson with \$20,000 in monthly sales will earn a 15% commission (\$3,000).

Because the staff gets paid once per month, Hal allows each employee to take up to \$1,500 per month in advance. When sales commissions are calculated, the amount of each employee's advanced pay is subtracted from the commission. If any salesperson's commission is less than the amount of this advance, he or she must reimburse Hal for the difference.

From *Starting Out with Java: From Control Structures through Objects*, Sixth Edition. Tony Gaddis. Copyright © 2016 by Pearson Education, Inc. All rights reserved.

**CS1-1**

**CS1-2** Case Study 1 Calculating Sales Commission

Here are two examples: Beverly and John have \$21,400 and \$12,600 in sales, respectively. Beverly’s commission is \$3,210 and John’s commission is \$1,260. Both Beverly and John took \$1,500 in advance pay. At the end of the month, Beverly gets a check for \$1,710, but John must pay \$240 back to Hal.

Now we will examine a program that eases the task of calculating the end-of-month commission.

**Program Design**

The program must perform the following general steps:

- 1. Ask the user to enter the salesperson’s monthly sales.
- 2. Ask the user to enter the amount of advanced pay.
- 3. Use the amount of monthly sales to determine the commission rate.
- 4. Calculate the commission.
- 5. Calculate the salesperson’s pay by subtracting the amount of advanced pay from the commission. If the amount is negative, the salesperson must reimburse the company.

Table CS1-2 lists and describes the program’s variables.

**Table CS1-2** Variables

Variable	Description
sales	A double variable to hold a salesperson’s total monthly sales.
rate	A double variable to hold the salesperson’s commission rate.
commission	A double variable to hold the commission.
advance	A double variable to hold the amount of advanced pay the salesperson has drawn.
Pay	A double variable to hold the salesperson’s amount of gross pay.

Programmers often write parts of a program in pseudocode before writing the actual Java code. Recall from Chapter 1 that *pseudocode* is a cross between human language and a programming language. It is an informal way of writing down each operation that a program must perform, and is especially helpful when designing an algorithm. Although the computer can’t understand pseudocode, most programmers find pseudocode helpful in creating a model of a program. When the pseudocode model is complete, it can be converted to actual code. Here is a pseudocode model for the sales commission program:

```
Display “Enter the amount of monthly sales”.
Read sales.
Display “Enter the amount of advanced pay”.
Read advance.
If sales is less than 10,000
    Store 0.05 in rate.
else if sales is less than 15,000
    Store 0.1 in rate.
```

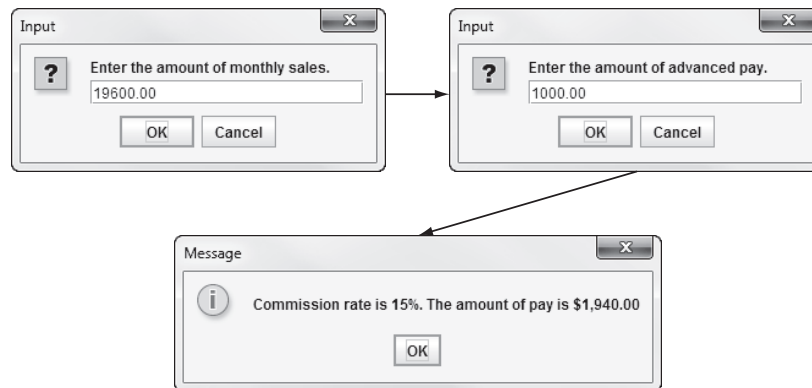
```

else if sales is less than 18,000
    Store 0.12 in rate.
else if sales is less than 22,000
    Store 0.15 in rate.
else
    Store 0.16 in rate.
commission = sales * rate.
pay = commission - advance.
Display pay.

```

Code Listing CS1-1 shows the actual program. Figure CS1-1 shows an example of interaction with the program.

**Figure CS1-1** Interaction with the `HalsCommission` program



**Code Listing CS1-1** (`HalsCommission.java`)

```

1  import javax.swing.JOptionPane; // Needed for JOptionPane
2
3  /**
4   This program calculates a salesperson's gross
5   pay at Hal's Computer Emporium.
6   */
7
8  public class HalsCommission
9  {
10     public static void main(String[] args)
11     {
12         String input;        // To hold the user's input
13         double sales;        // Monthly sales
14         double rate;         // Commission rate
15         double commission;   // Amount of commission

```

#### CS1-4 Case Study 1 Calculating Sales Commission

```
16      double advance;      // Advanced pay
17      double pay;          // Salesperson's pay
18
19      // Get the monthly sales.
20      input = JOptionPane.showInputDialog("Enter the amount " +
21                                         "of monthly sales.");
22      sales = Double.parseDouble(input);
23
24      // Get the advanced pay.
25      input = JOptionPane.showInputDialog("Enter the amount " +
26                                         "of advanced pay.");
27      advance = Double.parseDouble(input);
28
29      // Determine the rate of commission.
30      if (sales < 10000)
31          rate = 0.05;      // 5% commission rate
32      else if (sales < 15000)
33          rate = 0.1;       // 10% commission rate
34      else if (sales < 18000)
35          rate = 0.12;      // 12% commission rate
36      else if (sales < 22000)
37          rate = 0.15;      // 15% commission rate
38      else
39          rate = 0.16;      // 16% commission rate
40
41      // Calculate the amount of commission.
42      commission = rate * sales;
43
44      // Calculate the salesperson's pay.
45      pay = commission - advance;
46
47      // Display the salesperson's commission rate and pay.
48      JOptionPane.showMessageDialog(null,
49      String.format("Commission rate is %.2f. The amount of pay is $%,.2f",
50                  rate, pay));
51      System.exit(0);
52  }
53 }
```