

Programming Fundamentals II

Lab 7&8: Assignment

Assignment Guidelines:

- Feel free to use any IDE for completing this laboratory assignment.
- This lab assignment comprises **two problems** that necessitate resolution.
- Please ascertain that your code adheres to proper formatting and is adequately commented.
- Submit the code for each problem under its corresponding exercise number in a .java.(e.g Exercise1.zip).
- Make sure to submit the PDF file that contains the **list of errors** for **Exercise 1 (Debugging)**.

Note: *If you have any inquiries, please feel free to reach out to us via the discussion platform accessible to participants of this lab.*

Exercise 1: Debugging

The program in this section does not run properly. Fix all the compilation errors, so that the program will compile successfully. Once the program compiles, compare the output to the sample output, and eliminate any logic errors that exist. The sample output demonstrates what the program's output should be once the program's code is corrected.

Broken Code:

```
1 // Debugging Problem Chapter 7: Sales2.java
2 // Program totals sales for salespeople and products.
3 import java.util.Scanner;
4
5 public class Sales2
6 {
7     public void calculateSales()
8     {
9         Scanner input = new Scanner( System.in );
10        // sales array holds data on number of each product sold
11        // by each salesman
12        double sales = new double[ 5 ][ 4 ];
13
14        System.out.print( "Enter sales person number (-1 to end): " );
15        int person = input.nextInt();
16
17        while ( person != -1 )
18        {
19            System.out.print( "Enter product number: " );
20            int product = input.nextInt();
21            System.out.print( "Enter sales amount: " );
22            double amount = input.nextDouble();
23
24            // error-check the input
25            if ( person < 1 && person > 5 &&
26                product >= 1 && product < 6 && amount >= 0 )
27                sales[ product - 1 ][ person - 1 ] += amount;
28            else
29                System.out.println( "Invalid input!" );
30        }
```

```

31 System.out.print( "Enter sales person number (-1 to end): " );
32 person = input.nextInt();
33 } // end while
34
35 // total for each salesperson
36 double salesPersonTotal[][] = new double[ 4 ];
37
38 // display the table
39 for ( int column = 0; column < 4; column++ )
40 salesPersonTotal[ column ][ row ] = 0;
41
42 System.out.printf( "%7s%14s%14s%14s%14s%10s\n",
43 "Product", "Salesperson 1", "Salesperson 2",
44 "Salesperson 3", "Salesperson 4", "Total" );
45
46 // for each column of each row, print the appropriate
47 // value representing a person's sales of a product
48 for ( int row = 0; row < 5; row++ )
49 {
50 double productTotal = 0.0;
51 System.out.printf( "%7d", ( row + 1 ) );
52
53 for ( int column = 0; column < 4; column++ ) {
54 System.out.printf( "%14.2f", sales[ column ][ row ] );
55 productTotal += sales[ column ][ row ];
56 salesPersonTotal[ column ] += sales[ column ][ row ];
57 } // end for
58
59 System.out.printf( "%10.2f\n", productTotal );
60 } // end for
61
62 System.out.printf( "%7s", "Total" );
63
64 for ( int column = 0; column < 4; column++ )
65 System.out.printf( "%14.2f", salesPersonTotal[ column ] );
66
67 System.out.println();
68 } // end method calculateSales
69 } // end class Sales2

```

Fig. L 11.1 | Sales2.java.

```

1 // Debugging Problem Chapter 7: Sales2Test.java
2 // Test application for class Sales2
3 public class Sales2Test
4 {
5     public static void main( String args[] )
6     {
7         Sales2 application = new Sales2();
8         application.calculateSales();
9     } // end main
10 } // end class Sales2Test

```

Fig. L 11.2 | Sales2Test.java

Sample Output

```

Enter sales person number (-1 to end): 1
Enter product number: 4
Enter sales amount: 1082
Enter sales person number (-1 to end): 2
Enter product number: 3
Enter sales amount: 998
Enter sales person number (-1 to end): 3
Enter product number: 1
Enter sales amount: 678
Enter sales person number (-1 to end): 4
Enter product number: 1
Enter sales amount: 1554
Enter sales person number (-1 to end): -1
Product Salesperson 1 Salesperson 2 Salesperson 3 Salesperson 4      Total
1          0.00          0.00          678.00          1554.00      2232.00
2          0.00          0.00          0.00          0.00          0.00
3          0.00          998.00          0.00          0.00          998.00
4         1082.00          0.00          0.00          0.00         1082.00
5          0.00          0.00          0.00          0.00          0.00
Total         1082.00          998.00          678.00          1554.00

```

Make sure to write the complete **List of Errors in the following format:**

- Line (line number) — The **Error (Explain the error)** and **Solution**.

Make sure to attach the working code (the complete code with the two classes) after you fix all the errors.

Exercise 2: (Algebra: multiply two matrices)

Write a method to multiply two matrices. The header of the method is:

```
public static double[][]  
multiplyMatrix(double[][] a, double[][] b)
```

To multiply matrix **a** by matrix **b**, the number of columns in **a** must be the same as the number of rows in **b**, and the two matrices must have elements of the same or compatible types. Let **c** be the result of the multiplication. Assume the column size of matrix **a** is **n**. Each element **c_{ij}** is **a_{i1} * b_{1j} + a_{i2} * b_{2j} + ... + a_{in} * b_{nj}**.

For example, for two 3 * 3 matrices **a** and **b**, **c** is

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

where $c_{ij} = a_{i1} * b_{1j} + a_{i2} * b_{2j} + a_{i3} * b_{3j}$.

Write a test program that prompts the user to enter two 3 * 3 matrices and displays their product. Here is a sample run:

```
Enter matrix1: 1 2 3 4 5 6 7 8 9 ↵ Enter  
Enter matrix2: 0 2 4 1 4.5 2.2 1.1 4.3 5.2 ↵ Enter  
The multiplication of the matrices is  
1 2 3      0 2.0 4.0      5.3 23.9 24  
4 5 6      * 1 4.5 2.2 = 11.6 56.3 58.2  
7 8 9      1.1 4.3 5.2    17.9 88.7 92.4
```



End of Lab!