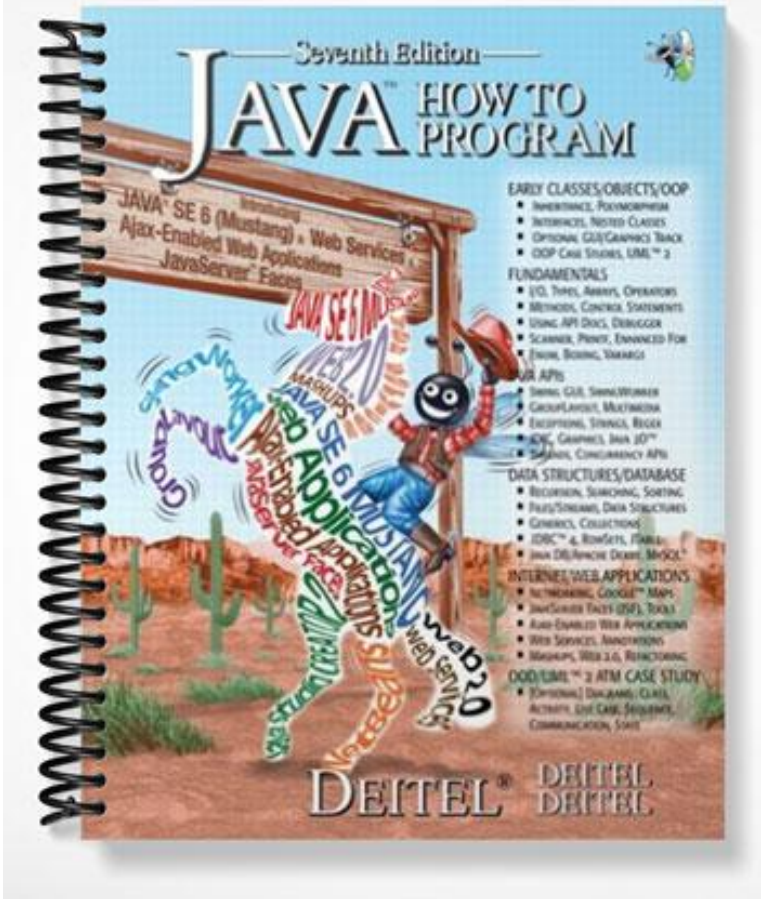
SOLUTIONS MANUAL



2

Introduction to Java Applications

What's in a name?
that which we call a rose
By any other name
would smell as sweet.
—William Shakespeare

When faced with a decision, I always ask, "What would be the most fun?"
—Peggy Walker

"Take some more tea," the March Hare said to Alice, very earnestly. "I've had nothing yet, "Alice replied in an offended tone: "so I can't take more." "You mean you can't take less," said the Hatter: "it's very easy to take more than nothing."
—Lewis Carroll

OBJECTIVES

In this chapter you will learn:

- To write simple Java applications.
- To use input and output statements.
- Java's primitive types.
- Basic memory concepts.
- To use arithmetic operators.
- The precedence of arithmetic operators.
- To write decision-making statements.
- To use relational and equality operators.

Self-Review Exercises

2.1 Fill in the blanks in each of the following statements:

a) A(n) _____ begins the body of every method, and a(n) _____ ends the body of every method.

ANS: left brace ({), right brace (}).

b) Every statement ends with a(n) _____.

ANS: semicolon (;).

c) The _____ statement is used to make decisions.

ANS: `if`.

d) _____ begins an end-of-line comment.

ANS: `//`.

e) _____, _____, _____ and _____ are called white space.

ANS: Blank lines, space characters, newline characters and tab characters.

f) _____ are reserved for use by Java.

ANS: Keywords.

g) Java applications begin execution at method _____.

ANS: `main`.

h) Methods _____, _____ and _____ display information in the command window.

ANS: `System.out.print`, `System.out.println` and `System.out.printf`.

2.2 State whether each of the following is *true* or *false*. If *false*, explain why.

a) Comments cause the computer to print the text after the `//` on the screen when the program executes.

ANS: False. Comments do not cause any action to be performed when the program executes. They are used to document programs and improve their readability.

b) All variables must be given a type when they are declared.

ANS: True.

c) Java considers the variables `number` and `NuMbEr` to be identical.

ANS: False. Java is case sensitive, so these variables are distinct.

d) The remainder operator (`%`) can be used only with integer operands.

ANS: False. The remainder operator can also be used with noninteger operands in Java.

e) The arithmetic operators `*`, `/`, `%`, `+` and `-` all have the same level of precedence.

ANS: False. The operators `*`, `/` and `%` are on the same level of precedence, and the operators `+` and `-` are on a lower level of precedence.

2.3 Write statements to accomplish each of the following tasks:

a) Declare variables `c`, `thisIsAVariable`, `q76354` and `number` to be of type `int`.

ANS:
```
int c, thisIsAVariable, q76354, number;
```
or
```
int c;
int thisIsAVariable;
int q76354;
int number;
```

b) Prompt the user to enter an integer.

ANS:
```
System.out.print( "Enter an integer: " );
```

c) Input an integer and assign the result to `int` variable `value`. Assume `Scanner` variable `input` can be used to read a value from the keyboard.

ANS:
```
value = input.nextInt();
```

d) If the variable number is not equal to 7, display "The variable number is not equal to 7".

ANS:
```
if ( number != 7 )
   System.out.println( "The variable number is not equal to 7" );
```

e) Print "This is a Java program" on one line in the command window.

ANS:
```
System.out.println( "This is a Java program" );
```

f) Print "This is a Java program" on two lines in the command window. The first line should end with Java. Use method System.out.println.

ANS:
```
System.out.println( "This is a Java\nprogram" );
```

g) Print "This is a Java program" on two lines in the command window. The first line should end with Java. Use method System.out.printf and two %s format specifiers.

ANS:
```
System.out.printf( "%s\n%s\n", "This is a Java", "program" );
```

2.4 Identify and correct the errors in each of the following statements:

a)
```
if ( c < 7 );
   System.out.println( "c is less than 7" );
```

ANS: Error: Semicolon after the right parenthesis of the condition (c < 7) in the if. Correction: Remove the semicolon after the right parenthesis. [*Note*: As a result, the output statement will execute regardless of whether the condition in the if is true.]

b)
```
if ( c => 7 )
   System.out.println( "c is equal to or greater than 7" );
```

ANS: Error: The relational operator => is incorrect. Correction: Change => to >=.

2.5 Write declarations, statements or comments that accomplish each of the following tasks:

a) State that a program will calculate the product of three integers.

ANS:
```
// Calculate the product of three integers
```

b) Create a Scanner that reads values from the standard input.

ANS:
```
Scanner input = new Scanner( System.in );
```

c) Declare the variables x, y, z and result to be of type int.

ANS:
```
int x, y, z, result;
```
or
```
int x;
int y;
int z;
int result;
```

d) Prompt the user to enter the first integer.

ANS:
```
System.out.print( "Enter first integer: " );
```

e) Read the first integer from the user and store it in the variable x.

ANS:
```
x = input.nextInt();
```

f) Prompt the user to enter the second integer.

ANS:
```
System.out.print( "Enter second integer: " );
```

g) Read the second integer from the user and store it in the variable y.

ANS:
```
y = input.nextInt();
```

h) Prompt the user to enter the third integer.

ANS:
```
System.out.print( "Enter third integer: " );
```

i) Read the third integer from the user and store it in the variable z.

ANS:
```
z = input.nextInt();
```

j) Compute the product of the three integers contained in variables x, y and z, and assign the result to the variable result.

ANS:
```
result = x * y * z;
```

k) Display the message "Product is" followed by the value of the variable result.

ANS:
```
System.out.printf( "Product is %d\n", result );
```

2.6 Using the statements you wrote in Exercise 2.5, write a complete program that calculates and prints the product of three integers.

ANS:

```
1  // Ex. 2.6: Product.java
2  // Calculate the product of three integers.
3  import java.util.Scanner; // program uses Scanner
4
5  public class Product
6  {
7     public static void main( String args[] )
8     {
9        // create Scanner to obtain input from command window
10       Scanner input = new Scanner( System.in );
11
12       int x; // first number input by user
13       int y; // second number input by user
14       int z; // third number input by user
15       int result; // product of numbers
16
17       System.out.print( "Enter first integer: " ); // prompt for input
18       x = input.nextInt(); // read first integer
19
20       System.out.print( "Enter second integer: " ); // prompt for input
21       y = input.nextInt(); // read second integer
22
23       System.out.print( "Enter third integer: " ); // prompt for input
24       z = input.nextInt(); // read third integer
25
26       result = x * y * z; // calculate product of numbers
27
28       System.out.printf( "Product is %d\n", result );
29
30    } // end method main
31
32 } // end class Product
```

```
Enter first integer: 10
Enter second integer: 20
Enter third integer: 30
Product is 6000
```

Exercises

2.7 Fill in the blanks in each of the following statements:

a) _____ are used to document a program and improve its readability.

ANS: Comments.

b) A decision can be made in a Java program with a(n) _____.

ANS: `if` statement.

c) Calculations are normally performed by _____ statements.

ANS: assignment statements.

d) The arithmetic operators with the same precedence as multiplication are _____ and _____.

ANS: division (/), remainder (%)

e) When parentheses in an arithmetic expression are nested, the _____ set of parentheses is evaluated first.

ANS: innermost.

f) A location in the computer's memory that may contain different values at various times throughout the execution of a program is called a(n) _____.

ANS: variable.

2.8 Write Java statements that accomplish each of the following tasks:

a) Display the message `"Enter an integer: "`, leaving the cursor on the same line.

ANS:
```
System.out.print( "Enter an integer: " );
```

b) Assign the product of variables `b` and `c` to variable `a`.

ANS:
```
a = b * c;
```

c) State that a program performs a sample payroll calculation (i.e., use text that helps to document a program).

ANS:
```
// This program performs a simple payroll calculation.
```

2.9 State whether each of the following is *true* or *false*. If *false*, explain why.

a) Java operators are evaluated from left to right.

ANS: False. Some operators (e.g., assignment, =) evaluate from right to left.

b) The following are all valid variable names: `_under_bar_`, `m928134`, `t5`, `j7`, `her_sales$`, `his_$account_total`, `a`, `b$`, `c`, `z` and `z2`.

ANS: True.

c) A valid Java arithmetic expression with no parentheses is evaluated from left to right.

ANS: False. The expression is evaluated according to operator precedence.

d) The following are all invalid variable names: `3g`, `87`, `67h2`, `h22` and `2h`.

ANS: False. Identifier `h22` is a valid variable name.

2.10 Assuming that `x = 2` and `y = 3`, what does each of the following statements display?

a)
```
System.out.printf( "x = %d\n", x );
```

ANS:
```
x = 2
```

b)
```
System.out.printf( "Value of %d + %d is %d\n", x, x, ( x + x ) );
```

ANS:
```
Value of 2 + 2 is 4
```

c)
```
System.out.printf( "x =" );
```

ANS:
```
x =
```

d)
```
System.out.printf( "%d = %d\n", ( x + y ), ( y + x ) );
```

ANS:
```
5 = 5
```

2.11 Which of the following Java statements contain variables whose values are modified?

a)
```
p = i + j + k + 7;
```

b)
```
System.out.println( "variables whose values are destroyed" );
```

c)
```
System.out.println( "a = 5" );
```

d)
```
value = input.nextInt();
```

ANS: (a), (d).

2.12 Given that $y = ax^3 + 7$, which of the following are correct Java statements for this equation?

a)
```
y = a * x * x * x + 7;
```

b)
```
y = a * x * x * ( x + 7 );
```

c)
```
y = ( a * x ) * x * ( x + 7 );
```

d)
```
y = ( a * x ) * x * x + 7;
```

e)
```
y = a * ( x * x * x ) + 7;
```

f) `y = a * x * ( x * x + 7 );`
ANS: (a), (d), (e)

2.13 State the order of evaluation of the operators in each of the following Java statements, and show the value of x after each statement is performed:

a) `x = 7 + 3 * 6 / 2 - 1;`
ANS: *, /, +, -; Value of x is 15.
b) `x = 2 % 2 + 2 * 2 - 2 / 2;`
ANS: %, *, /, +, -; Value of x is 3.
c) `x = ( 3 * 9 * ( 3 + ( 9 * 3 / ( 3 ) ) ) );`
ANS:
```
x = ( 3 * 9 * ( 3 + ( 9 * 3 / ( 3 ) ) ) );
          4   5     3     1   2
```
Value of x is 324.

2.14 Write an application that displays the numbers 1 to 4 on the same line, with each pair of adjacent numbers separated by one space. Write the program using the following techniques:

a) Use one `System.out.println` statement.
b) Use four `System.out.print` statements.
c) Use one `System.out.printf` statement.
ANS:

```
1  // Exercise 2.14 Solution: Printing.java
2  // Prints the numbers 1 through 4 several ways.
3
4  public class Printing
5  {
6     public static void main( String args[] )
7     {
8        System.out.print( "Part (a): " );
9
10       // one System.out.println statement
11       System.out.println( "1 2 3 4" );
12
13       System.out.print( "Part (b): " );
14
15       // four System.out.print statements
16       System.out.print( "1 " );
17       System.out.print( "2 " );
18       System.out.print( "3 " );
19       System.out.print( "4\n" );
20
21       System.out.print( "Part (c): " );
22
23       // one System.out.printf statement
24       System.out.printf( "%d %d %d %d\n", 1, 2, 3, 4 );
25    } // end main
26 } // end class Printing
```

```
Part (a): 1 2 3 4
Part (b): 1 2 3 4
Part (c): 1 2 3 4
```

2.15 Write an application that asks the user to enter two integers, obtains them from the user and prints their sum, product, difference and quotient (division). Use the techniques shown in Fig. 2.7.

ANS:

```
1  // Exercise 2.15 Solution: Calculate.java
2  // Prints the sum, product, difference and quotient of two numbers.
3  import java.util.Scanner;
4
5  public class Calculate
6  {
7     public static void main( String args[] )
8     {
9        Scanner input = new Scanner( System.in );
10
11       int number1; // first number
12       int number2; // second number
13
14       System.out.print( "Enter first integer: " ); // prompt for input
15       number1 = input.nextInt(); // read first integer
16
17       System.out.print( "Enter second integer: " ); // prompt for input
18       number2 = input.nextInt(); // read second integer
19
20       // display results
21       System.out.printf( "\nSum is %d\n", ( number1 + number2 ) );
22       System.out.printf( "Product is %d\n", ( number1 * number2 ) );
23       System.out.printf( "Difference is %d\n", ( number1 - number2 ) );
24       System.out.printf( "Quotient is %d\n", ( number1 / number2 ) );
25    } // end main
26 } // end class Calculate
```

```
Enter first integer: 45
Enter second integer: 5

Sum is 50
Product is 225
Difference is 40
Quotient is 9
```

2.16 Write an application that asks the user to enter two integers, obtains them from the user and displays the larger number followed by the words "`is larger`". If the numbers are equal, print the message "`These numbers are equal.`" Use the techniques shown in Fig. 2.15.

ANS:

```
1  // Exercise 2.16 Solution: Larger.java
2  // Program that determines the larger of two numbers.
3  import java.util.Scanner;
4
5  public class Larger
6  {
```

```
7      public static void main( String args[] )
8      {
9         Scanner input = new Scanner( System.in );
10
11        int number1; // first number to compare
12        int number2; // second number to compare
13
14        System.out.print( "Enter first integer: " ); // prompt for input
15        number1 = input.nextInt(); // read first number
16
17        System.out.print( "Enter second integer: " ); // prompt for input
18        number2 = input.nextInt(); // read second number
19
20        if ( number1 > number2 )
21           System.out.printf( "%d is larger\n", number1 );
22
23        if ( number1 < number2 )
24           System.out.printf( "%d is larger\n", number2 );
25
26        if ( number1 == number2 )
27           System.out.println( "These numbers are equal\n" );
28     } // end main
29  } // end class Larger
```

```
Enter first integer: 12
Enter second integer: 10
12 is larger
```

```
Enter first integer: 10
Enter second integer: 12
12 is larger
```

```
Enter first integer: 7
Enter second integer: 7
These numbers are equal
```

2.17 Write an application that inputs three integers from the user and displays the sum, average, product, smallest and largest of the numbers. Use the techniques shown in Fig. 2.15. [*Note*: The calculation of the average in this exercise should result in an integer representation of the average. So if the sum of the values is 7, the average should be 2, not 2.3333….]

ANS:

```
1  // Exercise 2.17 Solution: Calculate2.java
2  // Make simple calculations on three integers.
3  import java.util.Scanner;
4
5  public class Calculate2
6  {
```

```
7     public static void main( String args[] )
8     {
9        Scanner input = new Scanner( System.in );
10
11       int number1; // first number
12       int number2; // second number
13       int number3; // third number
14       int largest; // largest value
15       int smallest; // smallest value
16       int sum; // sum of numbers
17       int product; // product of numbers
18       int average; // average of numbers
19
20       System.out.print( "Enter first integer: " ); // prompt for input
21       number1 = input.nextInt(); // read first number
22       System.out.print( "Enter second integer: " ); // prompt for input
23       number2 = input.nextInt(); // read second number
24       System.out.print( "Enter third integer: " ); // prompt for input
25       number3 = input.nextInt(); // read third number
26
27       // determine largest value
28       largest = number1; // assume number1 is the largest
29
30       if ( number2 > largest ) // determine whether number2 is larger
31          largest = number2;
32
33       if ( number3 > largest ) // determine whether number3 is larger
34          largest = number3;
35
36       // determine smallest value
37       smallest = number1; // assume number1 is the smallest
38
39       if ( number2 < smallest ) // determine whether number2 is smallest
40          smallest = number2;
41
42       if ( number3 < smallest ) // determine whether number3 is smallest
43          smallest = number3;
44
45       // perform calculations
46       sum = number1 + number2 + number3;
47       product = number1 * number2 * number3;
48       average = sum / 3;
49
50       // print results
51       System.out.printf( "\nFor the numbers %d, %d and %d\n",
52          number1, number2, number3 );
53       System.out.printf( "Largest is %d\n", largest );
54       System.out.printf( "Smallest is %d\n", smallest );
55       System.out.printf( "Sum is %d\n", sum);
56       System.out.printf( "Product is %d\n", product );
57       System.out.printf( "Average is %d\n", average );
58    } // end main
59 } // end class Calculate2
```

```
Enter first integer: 10
Enter second integer: 20
Enter third integer: 30

For the numbers 10, 20 and 30
Largest is 30
Smallest is 10
Sum is 60
Product is 6000
Average is 20
```

```
Enter first integer: 30
Enter second integer: 20
Enter third integer: 10

For the numbers 30, 20 and 10
Largest is 30
Smallest is 10
Sum is 60
Product is 6000
Average is 20
```

```
Enter first integer: 10
Enter second integer: 30
Enter third integer: 20

For the numbers 10, 30 and 20
Largest is 30
Smallest is 10
Sum is 60
Product is 6000
Average is 20
```

2.18 Write an application that displays a box, an oval, an arrow and a diamond using asterisks (*), as follows:

ANS:

```
1  // Exercise 2.18 Solution: Shapes.java
2  // Program draws four shapes to the command window.
```

```
3
4  public class Shapes
5  {
6     public static void main( String args[] )
7     {
8        System.out.println( "*********      ***        *         *     " );
9        System.out.println( "*        *   *     *    ***       * *    " );
10       System.out.println( "*        *  *       *  *****     *   *   " );
11       System.out.println( "*        *  *       *    *      *     * " );
12       System.out.println( "*        *  *       *    *     *       *" );
13       System.out.println( "*        *  *       *    *      *     * " );
14       System.out.println( "*        *  *       *    *       *   *  " );
15       System.out.println( "*        *   *     *     *        * *    " );
16       System.out.println( "*********      ***        *         *     " );
17    } // end main
18 } // end class Shapes
```

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

2.19 What does the following code print?

```
System.out.println( "*\n**\n***\n****\n*****" );
```

ANS:

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

2.20 What does the following code print?

```
System.out.println( "*" );
System.out.println( "***" );
System.out.println( "*****" );
System.out.println( "****" );
System.out.println( "**" );
```

ANS:

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

2.21 What does the following code print?

```
System.out.print( "*" );
System.out.print( "***" );
System.out.print( "*****" );
System.out.print( "****" );
System.out.println( "**" );
```

ANS:

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

2.22 What does the following code print?

```
System.out.print( "*" );
System.out.println( "***" );
System.out.println( "*****" );
System.out.print( "****" );
System.out.println( "**" );
```

ANS:

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

2.23 What does the following code print?

```
System.out.printf( "%s\n%s\n%s\n", "*", "***", "*****" );
```

ANS:

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

2.24 Write an application that reads five integers, determines and prints the largest and smallest integers in the group. Use only the programming techniques you learned in this chapter.

ANS:

```
// Exercise 2.24 Solution: LargeSmall.java
// Program calculates the largest and smallest
// of five integers entered one at a time.
import java.util.Scanner;

public class LargeSmall
{
   public static void main( String args[] )
   {
      Scanner input = new Scanner( System.in );

      // numbers to be entered
      int firstNumber;
      int secondNumber;
      int thirdNumber;
      int fourthNumber;
      int fifthNumber;

      // largest and smallest
      int largest;
      int smallest;

      System.out.print( "Enter first number: " ); // prompt for input
      firstNumber = input.nextInt(); // read first number

      // initially firstNumber is the smallest and the largest
      smallest = firstNumber;
      largest = firstNumber;

      System.out.print( "Enter second number: " ); // prompt for input
      secondNumber = input.nextInt(); // read second number

      // determine whether secondNumber is the smallest
      if ( secondNumber < smallest )
         smallest = secondNumber;

      // determine whether secondNumber is the largest
      if ( secondNumber > largest )
         largest = secondNumber;

      System.out.print( "Enter third number: " ); // prompt for input
      thirdNumber = input.nextInt(); // read third number

      // determine whether thirdNumber is the smallest
      if ( thirdNumber < smallest )
         smallest = thirdNumber;

      // determine whether thirdNumber is the largest
      if ( thirdNumber > largest )
         largest = thirdNumber;
```

```
51
52          System.out.print( "Enter fourth number: " ); // prompt for input
53          fourthNumber = input.nextInt(); // read fourth number
54
55          // determine whether fourthNumber is the smallest
56          if ( fourthNumber < smallest )
57             smallest = fourthNumber;
58
59          // determine whether fourthNumber is the largest
60          if ( fourthNumber > largest )
61             largest = fourthNumber;
62
63          System.out.print( "Enter fifth number: " ); // prompt for input
64          fifthNumber = input.nextInt(); // read fifth number
65
66          // determine whether fifthNumber is the smallest
67          if ( fifthNumber < smallest )
68             smallest = fifthNumber;
69
70          // determine whether fifthNumber is the largest
71          if ( fifthNumber > largest )
72             largest = fifthNumber;
73
74          // display results
75          System.out.printf( "Numbers input: %d %d %d %d %d\n\n", firstNumber,
76             secondNumber, thirdNumber, fourthNumber, fifthNumber );
77          System.out.printf( "Smallest number is: %d\n", smallest );
78          System.out.printf( "Largest number is: %d\n", largest );
79       } // end main
80    } // end class LargeSmall
```

```
Enter first number: 20
Enter second number: 10
Enter third number: 40
Enter fourth number: 50
Enter fifth number: 30
Numbers input: 20 10 40 50 30

Smallest number is: 10
Largest number is: 50
```

2.25 Write an application that reads an integer and determines and prints whether it is odd or even. [*Hint*: Use the remainder operator. An even number is a multiple of 2. Any multiple of 2 leaves a remainder of 0 when divided by 2.]

ANS:

```
1  // Exercise 2.25 Solution: OddEven.java
2  // Program that determines if a number is odd or even.
3  import java.util.Scanner;
4
5  public class OddEven
6  {
```

```
7      public static void main( String args[] )
8      {
9         Scanner input = new Scanner( System.in );
10
11        int number; // number
12
13        System.out.print( "Enter integer: " ); // prompt for input
14        number = input.nextInt(); // read number
15
16        if ( number % 2 == 0 )
17           System.out.println( "Number is even" );
18
19        if ( number % 2 != 0 )
20           System.out.println( "Number is odd" );
21     } // end main
22  } // end class OddEven
```

```
Enter integer: 17
Number is odd
```

```
Enter integer: 244
Number is even
```

2.26 Write an application that reads two integers, determines whether the first is a multiple of the second and prints the result. [*Hint*: Use the remainder operator.]

ANS:

```
1  // Exercise 2.26 Solution: Multiple.java
2  // Program determines if the first number entered is a multiple
3  // of the second number entered.
4  import java.util.Scanner;
5
6  public class Multiple
7  {
8     public static void main( String args[] )
9     {
10       Scanner input = new Scanner( System.in );
11
12       int firstNumber;
13       int secondNumber;
14
15       System.out.print( "Enter first number: " ); // prompt for input
16       firstNumber = input.nextInt(); // read first number
17
18       System.out.print( "Enter second number: " ); // prompt for input
19       secondNumber = input.nextInt(); // read second number
20
21       // determine whether firstNumber is a multiple of secondNumber
22       if ( firstNumber % secondNumber == 0 )
23          System.out.printf( "%d is a multiple of %d\n",
```

```
            firstNumber, secondNumber );

      if ( firstNumber % secondNumber != 0 )
         System.out.printf( "%d is not a multiple of %d\n",
            firstNumber, secondNumber );
   } // end main
} // end class Multiple
```

```
Enter first number: 10
Enter second number: 2
10 is a multiple of 2
```

```
Enter first number: 17
Enter second number: 3
17 is not a multiple of 3
```

2.27 Write an application that displays a checkerboard pattern, as follows:

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

ANS:

```
// Exercise 2.27 Solution: Checker.java
// Program that draws a checkerboard.

public class Checker
{
   public static void main( String args[] )
   {
      System.out.println( "* * * * * * * *" );
      System.out.println( " * * * * * * * *" );
      System.out.println( "* * * * * * * *" );
      System.out.println( " * * * * * * * *" );
      System.out.println( "* * * * * * * *" );
      System.out.println( " * * * * * * * *" );
      System.out.println( "* * * * * * * *" );
      System.out.println( " * * * * * * * *" );
   } // end main
} // end class Checker
```

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

2.28 Here's a peek ahead. In this chapter, you have learned about integers and the type `int`. Java can also represent floating-point numbers that contain decimal points, such as 3.14159. Write an application that inputs from the user the radius of a circle as an integer and prints the circle's diameter, circumference and area using the floating-point value 3.14159 for π. Use the techniques shown in Fig. 2.7. [*Note*: You may also use the predefined constant `Math.PI` for the value of π. This constant is more precise than the value 3.14159. Class `Math` is defined in package `java.lang`. Classes in that package are imported automatically, so you do not need to `import` class `Math` to use it.] Use the following formulas (*r* is the radius):

diameter = 2*r*
circumference = 2π*r*
area = πr^2

Do not store the results of each calculation in a variable. Rather, specify each calculation as the value that will be output in a `System.out.printf` statement. Note that the values produced by the circumference and area calculations are floating-point numbers. Such values can be output with the format specifier `%f` in a `System.out.printf` statement. You will learn more about floating-point numbers in Chapter 3.

ANS:

```
1  // Exercise 2.28 Solution: Circle.java
2  // Program that calculates area, circumference
3  // and diameter for a circle.
4  import java.util.Scanner;
5
6  public class Circle
7  {
8     public static void main( String args[] )
9     {
10       Scanner input = new Scanner( System.in );
11
12       int radius; // radius of circle
13
14       System.out.print( "Enter radius: " ); // prompt for input
15       radius = input.nextInt(); // read number
16
17       System.out.printf( "Diameter is %d\n", ( 2 * radius ) );
18       System.out.printf( "Area is %f\n", ( Math.PI * radius * radius ) );
19       System.out.printf( "Circumference is %f\n",
20          ( 2 * Math.PI * radius ) );
21    } // end main
22 } // end class Circle
```

```
Enter radius: 3
Diameter is 6
Area is 28.274334
Circumference is 18.849556
```

2.29 Here's another peek ahead. In this chapter, you have learned about integers and the type `int`. Java can also represent uppercase letters, lowercase letters and a considerable variety of special symbols. Every character has a corresponding integer representation. The set of characters a computer uses and the corresponding integer representations for those characters is called that computer's character set. You can indicate a character value in a program simply by enclosing that character in single quotes, as in `'A'`.

You can determine the integer equivalent of a character by preceding that character with `(int)`, as in

```
(int) 'A'
```

This form is called a cast operator. (You will learn about cast operators in Chapter 4.) The following statement outputs a character and its integer equivalent:

```
System.out.printf(
   "The character %c has the value %d\n", 'A', ( (int) 'A' ) );
```

When the preceding statement executes, it displays the character `A` and the value `65` (from the so-called Unicode® character set) as part of the string. Note that the format specifier `%c` is a placeholder for a character (in this case, the character `'A'`).

Using statements similar to the one shown earlier in this exercise, write an application that displays the integer equivalents of some uppercase letters, lowercase letters, digits and special symbols. Display the integer equivalents of the following: `A B C a b c 0 1 2 $ * + /` and the blank character.

ANS:

```
1  // Exercise 2.29 Solution: Display.java
2  // Program that prints a unicode character
3  // and its integer equivalent.
4
5  public class Display
6  {
7     public static void main( String args[] )
8     {
9        System.out.printf( "The character %c has the value %d\n",
10          'A', ( (int) 'A' ) );
11       System.out.printf( "The character %c has the value %d\n",
12          'B', ( (int) 'B' ) );
13       System.out.printf( "The character %c has the value %d\n",
14          'C', ( (int) 'C' ) );
15       System.out.printf( "The character %c has the value %d\n",
16          'a', ( (int) 'a' ) );
17       System.out.printf( "The character %c has the value %d\n",
18          'b', ( (int) 'b' ) );
19       System.out.printf( "The character %c has the value %d\n",
20          'c', ( (int) 'c' ) );
21       System.out.printf( "The character %c has the value %d\n",
```

```
            '0', ( (int) '0' ) );
         System.out.printf( "The character %c has the value %d\n",
            '1', ( (int) '1' ) );
         System.out.printf( "The character %c has the value %d\n",
            '2', ( (int) '2' ) );
         System.out.printf( "The character %c has the value %d\n",
            '$', ( (int) '$' ) );
         System.out.printf( "The character %c has the value %d\n",
            '*', ( (int) '*' ) );
         System.out.printf( "The character %c has the value %d\n",
            '+', ( (int) '+' ) );
         System.out.printf( "The character %c has the value %d\n",
            '/', ( (int) '/' ) );
         System.out.printf( "The character %c has the value %d\n",
            ' ', ( (int) ' ' ) );
      } // end main
   } // end class Display
```

```
The character A has the value 65
The character B has the value 66
The character C has the value 67
The character a has the value 97
The character b has the value 98
The character c has the value 99
The character 0 has the value 48
The character 1 has the value 49
The character 2 has the value 50
The character $ has the value 36
The character * has the value 42
The character + has the value 43
The character / has the value 47
The character   has the value 32
```

2.30 Write an application that inputs one number consisting of five digits from the user, separates the number into its individual digits and prints the digits separated from one another by three spaces each. For example, if the user types in the number `42339`, the program should print

```
4   2   3   3   9
```

Assume that the user enters the correct number of digits. What happens when you execute the program and type a number with more than five digits? What happens when you execute the program and type a number with fewer than five digits? [*Hint*: It is possible to do this exercise with the techniques you learned in this chapter. You will need to use both division and remainder operations to "pick off" each digit.]

ANS: The last two sample outputs show the results of entering integers with fewer than five digits and more than five digits, respectively.

```
1  // Exercise 2.30 Solution: Five.java
2  // Program breaks apart a five-digit number
3  import java.util.Scanner;
4
```

```
5  public class Five
6  {
7     public static void main( String args[] )
8     {
9        Scanner input = new Scanner( System.in );
10
11       int number; // number input by user
12       int digit1; // first digit
13       int digit2; // second digit
14       int digit3; // third digit
15       int digit4; // fourth digit
16       int digit5; // fifth digit
17
18       System.out.print( "Enter five-digit integer: " ); // prompt
19       number = input.nextInt(); // read number
20
21       // determine the five digits
22       digit1 = number / 10000;
23       digit2 = number % 10000 / 1000;
24       digit3 = number % 10000 % 1000 / 100;
25       digit4 = number % 10000 % 1000 % 100 / 10;
26       digit5 = number % 10000 % 1000 % 100 % 10;
27
28       // output results
29       System.out.printf( "Digits in %d are %d  %d  %d  %d  %d\n",
30          number, digit1, digit2, digit3, digit4, digit5 );
31    } // end main
32 } // end class Five
```

```
Enter five-digit integer: 12345
Digits in 12345 are 1  2  3  4  5
```

```
Enter five-digit integer: 123
Digits in 123 are 0  0  1  2  3
```

```
Enter five-digit integer: 1234567
Digits in 1234567 are 123  4  5  6  7
```

2.31 Using only the programming techniques you learned in this chapter, write an application that calculates the squares and cubes of the numbers from 0 to 10 and prints the resulting values in table format, as shown below. [*Note*: This program does not require any input from the user.]

```
number  square  cube
0       0       0
1       1       1
2       4       8
3       9       27
4       16      64
5       25      125
6       36      216
7       49      343
8       64      512
9       81      729
10      100     1000
```

ANS:

```
1  // Exercise 2.31 Solution: Numbers.java
2  // Program prints a table of squares and cubes of numbers from 0 to 10.
3
4  public class Numbers
5  {
6     public static void main( String args[] )
7     {
8        // print a header for the table
9        System.out.printf( "%s\t%s\t%s\n", "number", "square", "cube" );
10
11       // print x, x squared and x cubed for each value
12       int x = 0;
13       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
14       x = 1;
15       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
16       x = 2;
17       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
18       x = 3;
19       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
20       x = 4;
21       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
22       x = 5;
23       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
24       x = 6;
25       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
26       x = 7;
27       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
28       x = 8;
29       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
30       x = 9;
31       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
32       x = 10;
33       System.out.printf( "%d\t%d\t%d\n", x, ( x * x ), ( x * x * x ) );
34    } // end main
35 } // end class Numbers
```

```
number  square  cube
0       0       0
1       1       1
2       4       8
3       9       27
4       16      64
5       25      125
6       36      216
7       49      343
8       64      512
9       81      729
10      100     1000
```

2.32 Write a program that inputs five numbers and determines and prints the number of negative numbers input, the number of positive numbers input and the number of zeros input.

ANS:

```
1  // Exercise 2.32 Solution: Tally.java
2  // Program accepts five numbers as input and prints a tally of the
3  // number of negatives, positives and zeros.
4  import java.util.Scanner;
5
6  public class Tally
7  {
8     public static void main( String args[] )
9     {
10       Scanner input = new Scanner( System.in );
11
12       int inputNumber;
13       int zeroTally;
14       int positiveTally;
15       int negativeTally;
16
17       // initialize counters
18       zeroTally = 0;
19       positiveTally = 0;
20       negativeTally = 0;
21
22       System.out.print( "Enter first integer: " ); // prompt for input
23       inputNumber = input.nextInt(); // read first number
24
25       if ( inputNumber == 0 )
26          zeroTally = zeroTally + 1;
27
28       if ( inputNumber < 0 )
29          negativeTally = negativeTally + 1;
30
31       if ( inputNumber > 0 )
32          positiveTally = positiveTally + 1;
33
34       System.out.print( "Enter second integer: " ); // prompt for input
35       inputNumber = input.nextInt(); // read second number
36
```

```
37         if ( inputNumber == 0 )
38            zeroTally = zeroTally + 1;
39
40         if ( inputNumber < 0 )
41            negativeTally = negativeTally + 1;
42
43         if ( inputNumber > 0 )
44            positiveTally = positiveTally + 1;
45
46         System.out.print( "Enter third integer: " ); // prompt for input
47         inputNumber = input.nextInt(); // read third number
48
49         if ( inputNumber == 0 )
50            zeroTally = zeroTally + 1;
51
52         if ( inputNumber < 0 )
53            negativeTally = negativeTally + 1;
54
55         if ( inputNumber > 0 )
56            positiveTally = positiveTally + 1;
57
58         System.out.print( "Enter fourth integer: " ); // prompt for input
59         inputNumber = input.nextInt(); // read fourth number
60
61         if ( inputNumber == 0 )
62            zeroTally = zeroTally + 1;
63
64         if ( inputNumber < 0 )
65            negativeTally = negativeTally + 1;
66
67         if ( inputNumber > 0 )
68            positiveTally = positiveTally + 1;
69
70         System.out.print( "Enter fifth integer: " ); // prompt for input
71         inputNumber = input.nextInt(); // read fifth number
72
73         if ( inputNumber == 0 )
74            zeroTally = zeroTally + 1;
75
76         if ( inputNumber < 0 )
77            negativeTally = negativeTally + 1;
78
79         if ( inputNumber > 0 )
80            positiveTally = positiveTally + 1;
81
82         // create a string describing the results
83         System.out.printf( "\nThere are %d zeros\n", zeroTally );
84         System.out.printf( "There are %d positive numbers\n",
85            positiveTally );
86         System.out.printf( "There are %d negative numbers\n",
87            negativeTally );
88      } // end main
89   } // end class Tally
```

```
Enter first integer: 0
Enter second integer: -7
Enter third integer: 3
Enter fourth integer: 13
Enter fifth integer: 5

There are 1 zeros
There are 3 positive numbers
There are 1 negative numbers
```