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

17. **2.23** What does the following code print?

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

18. **2.24** ***(Largest and Smallest Integers)*** Write an application that reads five integers and determines and prints the largest and smallest integers in the group. Use only the programming techniques you learned in this chapter.

19. **2.25** ***(Odd or Even)*** Write an application that reads an integer and determines and prints whether it's 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.]

20. **2.26** ***(Multiples)*** 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.]

21. **2.27** ***(Checkerboard Pattern of Asterisks)*** Write an application that displays a checkerboard pattern, as follows:

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

22. **2.28** ***(Diameter, Circumference and Area of a Circle)*** Here's a peek ahead. In this chapter, you 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):

$$diameter = 2r$$
$$circumference = 2\pi r$$
$$area = \pi r^2$$

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. 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'll learn more about floating-point numbers in Chapter 3 .

23. **2.29** ***(Integer Value of a Character)*** Here's another peek ahead. In this chapter, you 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 together with 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 a character's integer equivalent by preceding that character with `(int)`, as in

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

    An operator of this form is called a cast operator. (You'll 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 Unicode® character set) as part of the string. 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.

24. **2.30** ***(Separating the Digits in an Integer)*** 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 enter a number with more than five digits? What happens when you enter a number with fewer than five digits? [*Hint:* It's possible to do this exercise with the techniques you learned in this chapter. You'll need to use both division and remainder operations to "pick off " each digit.]

25. **2.31** ***(Table of Squares and Cubes)*** 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.

```
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
```

26. **2.32** ***(Negative, Positive and Zero Values)*** 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.