4. Trace the evaluation of the following expressions, and give their resulting values:

a. `4.0 / 2 * 9 / 2`

b. `2.5 * 2 + 8 / 5.0 + 10 / 3`

c. `12 / 7 * 4.4 * 2 / 4`

d. `4 * 3 / 8 + 2.5 * 2`

e. `(5 * 7.0 / 2 – 2.5) / 5 * 2`

f. `41 % 7 * 3 / 5 + 5 / 2 * 2.5`

g. `10.0 / 2 / 4`

h. `8 / 5 + 13 / 2 / 3.0`

i. `(2.5 + 3.5) / 2`

j. `9 / 4 * 2.0 – 5 / 4`

k. `9 / 2.0 + 7 / 3 – 3.0 / 2`

l. `813 % 100 / 3 + 2.4`

m. `27 / 2 / 2.0 * (4.3 + 1.7) – 8 / 3`

n. `53 / 5 / (0.6 + 1.4) / 2 + 13 / 2`

o. `2 * 3 / 4 * 2 / 4.0 + 4.5 – 1`

p. `89 % 10 / 4 * 2.0 / 5 + (1.5 + 1.0 / 2) * 2`

5. Trace the evaluation of the following expressions, and give their resulting values:

a. `2 + 2 + 3 + 4`

b. `"2 + 2" + 3 + 4`

c. `2 + " 2 + 3 " + 4`

d. `3 + 4 + " 2 + 2"`

e. `"2 + 2 " + (3 + 4)`

f. `"(2 + 2) " + (3 + 4)`

g. `"hello 34 " + 2 * 4`

h. `2 + "(int) 2.0" + 2 * 2 + 2`

i. `4 + 1 + 9 + "." + (-3 + 10) + 11 / 3`

j. `8 + 6 * –2 + 4 + "0" + (2 + 5)`

k. `1 + 1 + "8 – 2" + (8 – 2) + 1 + 1`

l. `5 + 2 + "(1 + 1)" + 4 + 2 * 3`

m. `"1" + 2 + 3 + "4" + 5 * 6 + "7" + (8 + 9)`

### Section 2.2: Variables

6. Which of the following choices is the correct syntax for declaring a real number variable named `grade` and initializing its value to `4.0`?

a. `int grade : 4.0;`

b. `grade = double 4.0;`

c. `double grade = 4.0;`

d. `grade = 4;`

e. `4.0 = grade;`

7. Imagine you are writing a personal fitness program that stores the user's age, gender, height (in feet or meters), and weight (to the nearest pound or kilogram). Declare variables with the appropriate names and types to hold this information.

**15.** Rewrite the code from the previous exercise to be shorter, by declaring the variables together and by using the speci assignment operators (e.g., += , −=, *=, and /=) as appropriate.

**16.** What are the values of i, j, and k after the following statements?

```
int i = 2;
int j = 3;
int k = 4;
int x = i + j + k;


i = x - i - j;
j = x - j - k;
k = x - i - k;
```

**17.** What is the output from the following code?

```
int max;
int min = 10;
max = 17 — 4 / 10;
max = max + 6;
min = max — min;
System.out.println(max * 2);
System.out.println(max + min);
System.out.println(max);
System.out.println(min);
```

**18.** Suppose you have a real number variable *x*. Write a Java expression that computes the following value *y* while usin the * operator only four times:

$y = 12.3x^4 - 9.1x^3 + 19.3x^2 - 4.6x + 34.2$

**19.** The following program redundantly repeats the same expressions many times. Modify the program to remove all redundant expressions using variables of appropriate types.

```
public class ComputePay {
    public static void main(String[] args) {
        // Calculate pay at work based on hours worked each day
        System.out.println("My total hours worked:");
        System.out.println(4 + 5 + 8 + 4);

        System.out.println("My hourly salary:");
        System.out.println("$8.75");

        System.out.println("My total pay:");
        System.out.println((4 + 5 + 8 + 4) * 8.75);
```

to produce the following output:

```
2 times 1 = 2
2 times 2 = 4
2 times 3 = 6
2 times 4 = 8
```

22. Assume that you have a variable called `count` that will take on the values `1, 2, 3, 4,` and so on. You are going to formulate expressions in terms of `count` that will yield different sequences. For example, to get the sequence `2, 4, 6, 8, 10, 12, ...,` you would use the expression `(2 * count)`. Fill in the following table, indicating an expression that will generate each sequence.

| Sequence | Expression |
|---|---|
| a. 2, 4, 6, 8, 10, 12, . . . | |
| b. 4, 19, 34, 49, 64, 79, . . . | |
| c. 30, 20, 10, 0, 10, 20, . . . | |
| d. 7, 3, 1, 5, 9, 13, . . . | |
| e. 97, 94, 91, 88, 85, 82, . . . | |

23. Complete the code for the following `for` loop:

```
for (int i = 1; i <= 6; i++) {
    // your code here
}
```

so that it prints the following numbers, one per line:

```
-4
14
32
50
68
86
```

24. What is the output of the following `oddStuff` method?

```
public static void oddStuff() {
    int number = 4;
    for (int count = 1; count <= number; count++) {
        System.out.println(number);
        number = number / 2;
    }
}
```

31. What is the output of the following sequence of loops?

```
for (int i = 1; i <= 2; i++) {
    for (int j = 1; j <= 3; j++) {
        for (int k = 1; k <= 4; k++) {
            System.out.print("*");
        }
        System.out.print("!");
    }
    System.out.println();
}
```

32. What is the output of the following sequence of loops? Notice that the code is the same as that in the previous exercise, except that the placement of the braces has changed.

```
for (int i = 1; i <= 2; i++) {
    for (int j = 1; j <= 3; j++) {
        for (int k = 1; k <= 4; k++) {
            System.out.print("*");
        }
    }
    System.out.print("!");
    System.out.println();
}
```

33. What is the output of the following sequence of loops? Notice that the code is the same as that in the previous exercise, except that the placement of the braces has changed.

```
for (int i = 1; i <= 2; i++) {
    for (int j = 1; j <= 3; j++) {
        for (int k = 1; k <= 4; k++) {
            System.out.print("*");
            System.out.print("!");
        }
        System.out.println();
    }
}
```

### Section 2.4: Managing Complexity

34. Suppose that you are trying to write a program that produces the following output:

```
1 3 5 7 9 11 13 15 17 19 21
1 3 5 7 9 11
```

The following program is an attempt at a solution, but it contains four major errors. Identify them all.

```
public class BadNews {
    public static final int MAX_ODD = 21;

    public static void writeOdds() {
        // print each odd number
        for (int count = 1; count <= (MAX_ODD  2); count++) {
            System.out.print(count + " ");
            count = count + 2;
        }

        // print the last odd number
        System.out.print(count + 2);
    }

    public static void main(String[] args) {
        // write all odds up to 21
        writeOdds();

        // now, write all odds up to 11
        MAX_ODD = 11;
        writeOdds();
    }
}
```

**35.** What is the output of the following unknown method?

```
public class Strange {
    public static final int MAX = 5;

    public static void unknown() {
        int number = 0;

        for (int count = MAX; count >= 1; count--) {
            number += (count * count);
        }

        System.out.println("The result is: " + number);
    }

    public static void main(String[] args) {
        unknown();
    }
}
```

**36.** Suppose that you have a variable called `line` that will take on the values `1, 2, 3, 4,` and so on, and a class constant named `SIZE` that takes one of two values. You are going to formulate expressions in terms of `line` and `SIZE` that will yield different sequences of numbers of characters. Fill in the table below, indicating an expression that will generate each sequence.

| line value | constant SIZE value | Number of characters | Expression |
|---|---|---|---|
| a. 1, 2, 3, 4, 5, 6, ... | 1 | 4, 6, 8, 10, 12, 14, ... | |
| 1, 2, 3, 4, 5, 6, ... | 2 | 6, 8, 10, 12, 14, 16, ... | |
| b. 1, 2, 3, 4, 5, 6, ... | 3 | 13, 17, 21, 25, 29, 33, ... | |
| 1, 2, 3, 4, 5, 6, ... | 5 | 19, 23, 27, 31, 35, 39, ... | |
| c. 1, 2, 3, 4, 5, 6, ... | 4 | 10, 9, 8, 7, 6, 5, ... | |
| 1, 2, 3, 4, 5, 6, ... | 9 | 20, 19, 18, 17, 16, 15, ... | |

**37.** Write a table that determines the expressions for the number of each type of character on each of the 6 lines in the following output.

```
!!!!!!!!!!!!!!!!!!!!!!
\\!!!!!!!!!!!!!!!!!!//
\\\\!!!!!!!!!!!!!!////
\\\\\\!!!!!!!!!!//////
\\\\\\\\!!!!!!////////
\\\\\\\\\\!!//////////
```

**38.** Suppose that a program has been written that produces the output shown in the previous problem. Now the author wants the program to be scalable using a class constant called `SIZE`. The previous output used a constant height of 6, since there were 6 lines. The following is the output for a constant height of 4. Create a new table that shows the expressions for the character counts at this new size of 4, and compare these tables to figure out the expressions for any size using the `SIZE` constant.

```
!!!!!!!!!!!!!!
\\!!!!!!!!!!//
\\\\!!!!!!////
\\\\\\!!//////
```

## Exercises

**1.** In physics, a common useful equation for finding the position $s$ of a body in linear motion at a given time $t$, based on its initial position $s_0$, initial velocity $v_0$, and rate of acceleration $a$, is the following:

$$s = s_0 + v_0t + \frac{1}{2}at^2$$

Write code to declare variables for $s_0$, $v_0$, $a$, and $t$, and then write the code to compute $s$ on the basis of these values.

**2.** Write a `for` loop that produces the following output:

```
1 4 9 16 25 36 49 64 81 100
```

For added challenge, try to modify your code so that it does not need to use the * multiplication operator. (It can be done! Hint: Look at the differences between adjacent numbers.)

**3.** The Fibonacci numbers are a sequence of integers in which the first tw ing element is the sum of the two preceding elements. Th following:

7. Write a program that produces the following figure (which vaguely resembles the Seattle Space Needle) as its output using nested `for` loops. Use a class constant to make it possible to change the size of the figure (the following output uses a size of 4).

```
            ||
            ||
            ||
            ||
         __/||\__
      __/:::||:::\__
   __/::::::||::::::\__
__/:::::::::||:::::::::\__
|""""""""""""""""""""""""|
\_/\/\/\/\/\/\/\/\/\/\/\_/
  \_/\/\/\/\/\/\/\/\/\_/
    \_/\/\/\/\/\/\/\_/
      \_/\/\/\/\/\_/
            ||
            ||
            ||
            ||
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         __/||\__
      __/:::||:::\__
   __/::::::||::::::\__
__/:::::::::||:::::::::\__
|""""""""""""""""""""""""|
```