

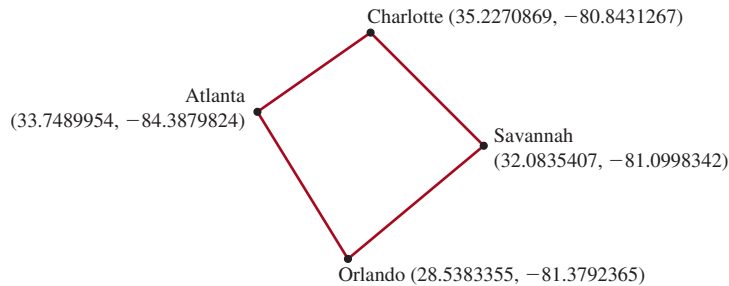
4.1 Introduction



The focus of this chapter is to introduce mathematical functions, characters, string objects, and use them to develop programs.

The preceding chapters introduced fundamental programming techniques and taught you how to write simple programs to solve basic problems using selection statements. This chapter introduces methods for performing common mathematical operations. You will learn how to create custom methods in Chapter 6.

Suppose you need to estimate the area enclosed by four cities, given the GPS locations (latitude and longitude) of these cities, as shown in the following diagram. How would you write a program to solve this problem? You will be able to write such a program in this chapter.



Because strings are frequently used in programming, it is beneficial to introduce strings early so that you can begin to use them to develop useful programs. This chapter also gives a brief introduction to string objects; you will learn more on objects and strings in Chapters 9 and 10.

4.2 Common Mathematical Functions



*Java provides many useful methods in the **Math** class for performing common mathematical functions.*

A method is a group of statements that performs a specific task. You have already used the `pow(a, b)` method to compute a^b in Section 2.9.4, Exponent Operations and the `random()` method for generating a random number in Section 3.7. This section introduces other useful methods in the **Math** class. They can be categorized as *trigonometric methods*, *exponent methods*, and *service methods*. Service methods include the rounding, min, max, absolute, and random methods. In addition to methods, the **Math** class provides two useful **double** constants, **PI** and **E** (the base of natural logarithms). You can use these constants as `Math.PI` and `Math.E` in any program.

4.2.1 Trigonometric Methods

The **Math** class contains the following methods as listed in Table 4.1 for performing trigonometric functions:

The parameter for **sin**, **cos**, and **tan** is an angle in radians. The return value for **asin** and **atan** is an angle in radians in the range between $-\pi/2$ and $\pi/2$, and for **acos** is between 0 and π . One degree is equal to $\pi/180$ in radians, 90 degrees is equal to $\pi/2$ in radians, and 30 degrees is equal to $\pi/6$ in radians.

For example,

```
Math.toDegrees(Math.PI / 2) returns 90.0
Math.toRadians(30) returns 0.5236 (same as  $\pi/6$ )
Math.sin(0) returns 0.0
```



VideoNote

Introduce Math functions

problem

TABLE 4.1 Trigonometric Methods in the Math Class

Method	Description
<code>sin(radians)</code>	Returns the trigonometric sine of an angle in radians.
<code>cos(radians)</code>	Returns the trigonometric cosine of an angle in radians.
<code>tan(radians)</code>	Returns the trigonometric tangent of an angle in radians.
<code>toRadians(degree)</code>	Returns the angle in radians for the angle in degrees.
<code>toDegrees(radians)</code>	Returns the angle in degrees for the angle in radians.
<code>asin(a)</code>	Returns the angle in radians for the inverse of sine.
<code>acos(a)</code>	Returns the angle in radians for the inverse of cosine.
<code>atan(a)</code>	Returns the angle in radians for the inverse of tangent.

```

Math.sin(Math.toRadians(270)) returns -1.0
Math.sin(Math.PI / 6) returns 0.5
Math.sin(Math.PI / 2) returns 1.0
Math.cos(0) returns 1.0
Math.cos(Math.PI / 6) returns 0.866
Math.cos(Math.PI / 2) returns 0
Math.asin(0.5) returns 0.523598333 (same as  $\pi/6$ )
Math.acos(0.5) returns 1.0472 (same as  $\pi/3$ )
Math.atan(1.0) returns 0.785398 (same as  $\pi/4$ )

```

4.2.2 Exponent Methods

There are five methods related to exponents in the **Math** class as listed in Table 4.2.

TABLE 4.2 Exponent Methods in the Math Class

Method	Description
<code>exp(x)</code>	Returns e raised to power of x (e^x).
<code>log(x)</code>	Returns the natural logarithm of x ($\ln(x) = \log_e(x)$).
<code>log10(x)</code>	Returns the base 10 logarithm of x ($\log_{10}(x)$).
<code>pow(a, b)</code>	Returns a raised to the power of b (a^b).
<code>sqrt(x)</code>	Returns the square root of x (\sqrt{x}) for $x \geq 0$.

For example,

```

e3.5 is Math.exp(3.5), which returns 33.11545
ln(3.5) is Math.log(3.5), which returns 1.25276
log10(3.5) is Math.log10(3.5), which returns 0.544
23 is Math.pow(2, 3), which returns 8.0
32 is Math.pow(3, 2), which returns 9.0
4.52.5 is Math.pow(4.5, 2.5), which returns 42.9567
 $\sqrt{4}$  is Math.sqrt(4), which returns 2.0
 $\sqrt{10.5}$  is Math.sqrt(10.5), which returns 3.24

```

4.2.3 The Rounding Methods

The **Math** class contains four rounding methods as listed in Table 4.3.

TABLE 4.3 Rounding Methods in the Math Class

Method	Description
<code>ceil(x)</code>	x is rounded up to its nearest integer. This integer is returned as a double value.
<code>floor(x)</code>	x is rounded down to its nearest integer. This integer is returned as a double value.
<code>rint(x)</code>	x is rounded to its nearest integer. If x is equally close to two integers, the even one is returned as a double value.
<code>round(x)</code>	Returns <code>(int)Math.floor(x + 0.5)</code> if x is a float and returns <code>(long)Math.floor(x + 0.5)</code> if x is a double.

For example,

```
Math.ceil(2.1) returns 3.0
Math.ceil(2.0) returns 2.0
Math.ceil(-2.0) returns -2.0
Math.ceil(-2.1) returns -2.0
Math.floor(2.1) returns 2.0
Math.floor(2.0) returns 2.0
Math.floor(-2.0) returns -2.0
Math.floor(-2.1) returns -3.0
Math.rint(2.1) returns 2.0
Math.rint(-2.0) returns -2.0
Math.rint(-2.1) returns -2.0
Math.rint(2.5) returns 2.0
Math.rint(3.5) returns 4.0
Math.rint(-2.5) returns -2.0
Math.round(2.6f) returns 3 // Returns int
Math.round(2.0) returns 2 // Returns long
Math.round(-2.0f) returns -2 // Returns int
Math.round(-2.6) returns -3 // Returns long
Math.round(-2.4) returns -2 // Returns long
```

4.2.4 The min, max, and abs Methods

The `min` and `max` methods return the minimum and maximum numbers of two numbers (`int`, `long`, `float`, or `double`). For example, `max(4.4, 5.0)` returns `5.0`, and `min(3, 2)` returns `2`.

The `abs` method returns the absolute value of the number (`int`, `long`, `float`, or `double`). For example,

```
Math.max(2, 3) returns 3
Math.min(2.5, 4.6) returns 2.5
Math.max(Math.max(2.5, 4.6), Math.min(3, 5.6)) returns 4.6
Math.abs(-2) returns 2
Math.abs(-2.1) returns 2.1
```

4.2.5 The random Method

You used the `random()` method in the preceding chapter. This method generates a random `double` value greater than or equal to 0.0 and less than 1.0 (`0 <= Math.random() < 1.0`). You can use it to write a simple expression to generate random numbers in any range. For example,

```
(int) (Math.random() * 10);      → Return a random integer
                                between 0 and 9.
(50 + int) (Math.random() * 50); → Return a random integer
                                between 50 and 99.
```

In general,

```
a + (int) (Math.random() * b); → Return a random integer
                                between a and a+b-1.
```