APPENDIX F

More about the Math Class

The Java `Math` class is a collection of static methods for performing specific mathematical operations. This class is in the `java.lang` package, so there is no need for an `import` statement to use it.

In Chapter 2, you were introduced to the `Math.pow` method, which returns the value of a number raised to a power. Table F-1 describes several of the `Math` class's methods.

Table F-1 Several `Math` class methods

Method	Example Usage	Description
`abs`	`y = Math.abs(x);`	Returns the absolute value of the argument. This method can accept and return values of the `double`, `float`, `int`, and `long` data types.
`acos`	`y = Math.acos(x);`	Returns the arc-cosine of the argument. The argument should be the cosine of an angle. (The argument's value must be in the range from –1.0 through 1.0.) The return type and the argument are `doubles`.
`asin`	`y = Math.asin(x);`	Returns the arc-sine of the argument. The argument should be the sine of an angle. (The argument's value must be in the range from –1.0 through 1.0.) The return type and the argument are `doubles`.
`atan`	`y = Math.atan(x);`	Returns the arc-tangent of the argument. The argument should be the tangent of an angle. The return type and the argument are `doubles`.
`cbrt`	`y = Math.cbrt(x);`	Returns the cube root of the argument. The return type and argument are `doubles`. Note: This method was introduced in Java 5.

Method	Example Usage	Description
`ceil`	`y = Math.ceil(x);`	Returns the smallest number that is greater than or equal to the argument. The return type and the argument are `doubles`.
`cos`	`y = Math.cos(x);`	Returns the cosine of the argument. The argument should be an angle expressed in radians. The return type and the argument are `doubles`.
`exp`	`y = Math.exp(x);`	Computes the exponential function of the argument, which is e^x. The return type and the argument are `doubles`.
`floor`	`y = Math.floor(x);`	Returns the largest number that is less than or equal to the argument. The return type and the argument are `doubles`.
`hypot`	`c = Math.hypot(a, b);`	Returns the length of the hypotenuse of a right triangle. The arguments `a` and `b` are the lengths of the other two sides of the triangle. The return type and the argument are `doubles`. Note: This method was introduced in Java 5.
`log`	`y = Math.log(x);`	Returns the natural logarithm of the argument. The return type and the argument are `doubles`.
`pow`	`y = Math.pow(x, z);`	Returns the value of the first argument raised to the power of the second argument. The return type and the argument are `doubles`.
`round`	`y = Math.round(x);`	Returns the value of the argument, as an integer, rounded to the nearest whole number. The argument is expected to be a `double` or a `float`. If the argument is a `double`, the return type is `long`. If the argument is a `float`, the return type is `int`.
`sin`	`y = Math.sin(x);`	Returns the sine of the argument. The argument should be an angle expressed in radians. The return type and the argument are `doubles`.
`sqrt`	`y = Math.sqrt(x);`	Returns the square root of the argument. The return type and argument are `doubles`.
`tan`	`y = Math.tan(x);`	Returns the tangent of the argument. The argument should be an angle expressed in radians. The return type and the argument are `doubles`.
`toDegrees`	`y = Math.toDegrees(x);`	Accepts as an argument an angle in radians. The angle converted to degrees is returned. The argument and return values are both `doubles`.
`toRadians`	`y = Math.toRadians(x);`	Accepts as an argument an angle in degrees. The angle converted to radians is returned. The argument and return values are both `doubles`.

We will not cover all of these methods in depth, but let's take a closer look at some of them. The `Math.abs` method returns the absolute value of its argument. The following code segment shows an example of how it is used.

```
double x = -4.2, y = 4.2, a, b;
a = Math.abs(x);
b = Math.abs(y);
```

After this code executes, both the variables `a` and `b` will contain the value 4.2. The argument to the `Math.abs` method can be of the `double`, `float`, `int`, or `long` data types. This is because there are several overloaded versions of the method. The method's return value will be of the same data type as the argument.

Here is a program segment that demonstrates the `Math.sqrt` method, which returns the square root of a number:

```
num = 25.0;
s = Math.sqrt(num);
System.out.println("The square root of " + num + " is " + s);
```

The output of this program segment is

```
The square root of 25.0 is 5.0
```

The `Math.round` method returns the value of its argument rounded to the nearest whole number. The following code segment shows an example.

```
double x = 4.2, y = 4.8;
long a, b;
a = Math.round(x);
b = Math.round(y);
```

After this code executes, `a` will contain 4 and `b` will contain 5.

In addition to these and other methods, the `Math` class defines two static `final` variables, which are listed in Table F-2.

Table F-2 The `E` and `PI` constants

Method	Example Usage	Description
`E`	`y = x * Math.E;`	This is the mathematical constant known as e (for Euler's number). It is defined as 2.7182818284590452354.
`PI`	`area = Math.PI * radius * radius;`	This is the mathematical constant Pi, or π. It is defined as 3.14159265358979323849.

Both of these `final` variables are public, so you can access them directly from the class.