

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

4.2.1 Trigonometric Methods

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.

Examples:

`Math.toDegrees(Math.PI/2)` returns **90.0**
`Math.toRadians(30)` returns **0.5236** (same as $\pi/6$)
`Math.sin(0)` returns **0.0**
`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/6)` returns **0.0**
`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 **1.785398** (same as $\pi/4$)

4.2.2 Exponent Methods

Table 4.2 Exponent Methods in the Math Class

Method	Description
<code>exp(x)</code>	Returns e raised to the 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 $x \geq 0$.

Examples:

$e^{3.5}$ is `Math.exp(3.5)`, which returns **33.11545**
 $\ln(3.5)$ is `Math.log(3.5)`, which returns **1.25276**
 $\log_{10}(3.5)$ is `Math.log10(3.5)`, which returns **0.544**
 2^3 is `Math.pow(2,3)`, which returns **8.0**
 3^2 is `Math.pow(3,2)`, which returns **9.0**
 $4.5^{2.5}$ is `Math.pow(4.5,2.5)`, which returns **42.9587**

$\sqrt{4}$ 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

Table 4.3 Rounding Methods in the Math Class

Method	Description
<code>ceil(x)</code>	x is rounded up to its nearest integer. The integer is return as a double value.
<code>floor(x)</code>	x is rounded down to its nearest integer. The integer is return 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

Examples:

```
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(-2.1) returns -3.0
Math.rint(2.1) returns -2.0
Math.rint(-2.0) returns 2.0
Math.rint(2.5) 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, abs Methods

Table 4.4 Rounding Methods in the Math Class

Method	Description
<code>min(x,y)</code>	The minimum of x and y is returned. x and y may have types int , long , float or double .
<code>max(x,y)</code>	The max of x and y is returned. x and y may have types int , long , float or double .
<code>abs(x)</code>	The magnitude or absolute value of x is returned. x may have type int , long , float or double .

Examples:

```
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 Methods

Table 4.5 Rounding Methods in the Math Class

Method	Description
<code>random()</code>	A value x of type double in the range $0.0 \leq x < 1.0$ is returned

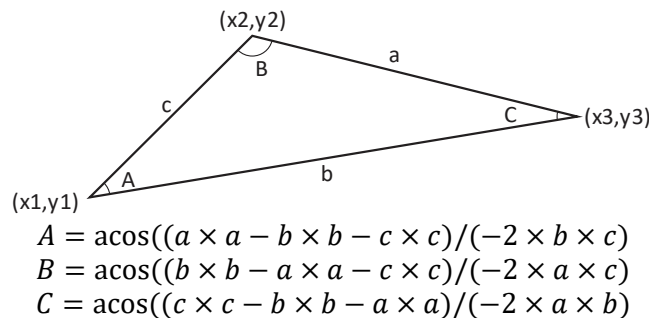
Examples

```
(int)(Math.random() * 10)           //Returns a random integer between 0 and 9
50 + (int)(Math.random() * 50)      //Returns a random integer between 50 and 99
```

In general,

```
a + Math.random() * b           //Returns a random number between a and a + b
                                //excluding a + b
```

4.2.6 Case Study: Computing Angles of a Triangle



Listing 4.1 ComputeAngles.java

```
1  import java.util.Scanner;
2
3  public class ComputeAngles {
4      public static void main(String[] args) {
5          Scanner input = new Scanner(System.in);
6
7          //Prompt the user to enter three points
8          System.out.print("Enter three points: ");
9          double x1 = input.nextDouble();
10         double y1 = input.nextDouble();
11         double x2 = input.nextDouble();
12         double y2 = input.nextDouble();
13         double x3 = input.nextDouble();
14         double y3 = input.nextDouble();
15
16         //Compute three sides
17         double a = Math.sqrt((x2 - x3) * (x2 - x3))
```

```
18         + (y2 - y3) * (y2 - y3));
19     double b = Math.sqrt((x1 - x3) * (x1 - x3)
20         + (y1 - y3) * (y1 - y3));
21     double c = Math.sqrt((x1 - x2) * (x1 - x2)
22         + (y1 - y2) * (y1 - y2));
23
24     //Compute three angles
25     double A = Math.toDegrees(Math.acos(a * a - b * b - c * c)
26         / (-2 * b * c));
27     double B = Math.toDegrees(Math.acos(b * b - a * a - c * c)
28         / (-2 * a * c));
29     double C = Math.toDegrees(Math.acos(c * c - b * b - a * a)
30         / (-2 * a * b));
31
32     //Display results
33     System.out.println("The three angles are " +
34         Math.round(A * 100) / 100.0 + " " +
35         Math.round(B * 100) / 100.0 + " " +
36         Math.round(C * 100) / 100.0);
37     }
38 }
```