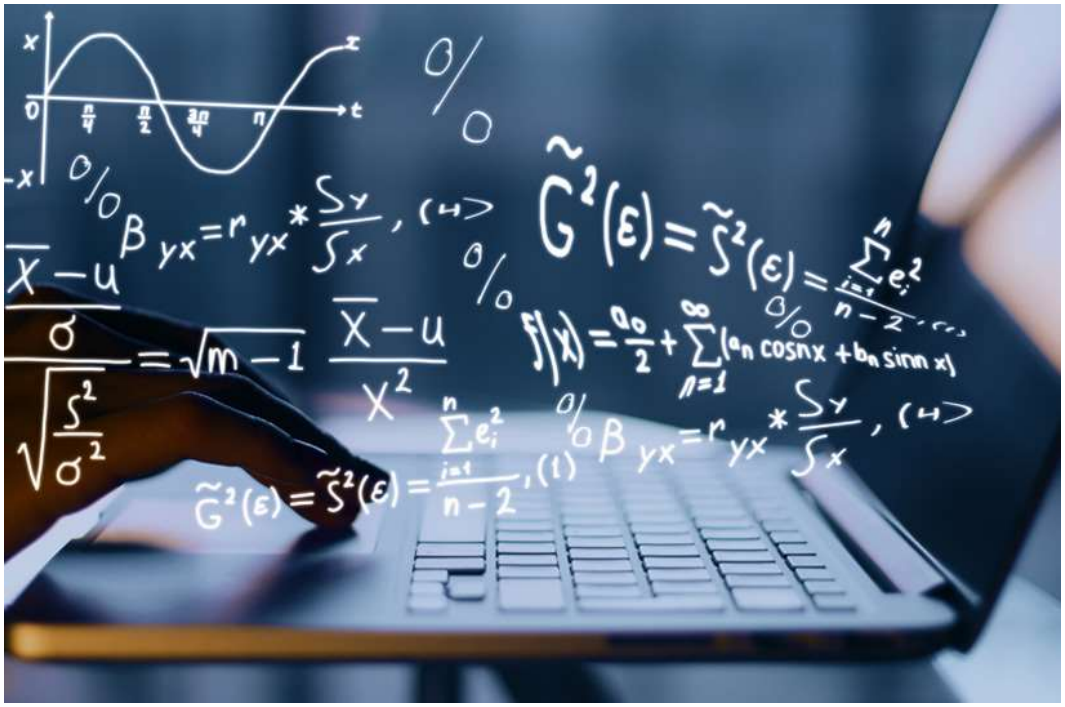

Math class

The java.lang.Math class contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions.

Following are the fields for java.lang.Math class –

- **static double E – This is the double value that is closer than any other to e, the base of the natural logarithms.**

- **static double PI – This is the double value that is closer than any other to pi, the ratio of the circumference of a circle to its diameter.**

Some of the important Math class functions

Java's `Math` class

Method name	Description
`Math.abs(`*value*`)`	absolute value
`Math.ceil(`*value*`)`	rounds up
`Math.floor(`*value*`)`	rounds down
`Math.log10(`*value*`)`	logarithm, base 10
`Math.max(`*value1*`, `*value2*`)`	larger of two values
`Math.min(`*value1*`, `*value2*`)`	smaller of two values
`Math.pow(`*base*`, `*exp*`)`	*base* to the *exp* power
`Math.random()`	random `double` between 0 and 1
`Math.round(`*value*`)`	nearest whole number
`Math.sqrt(`*value*`)`	square root

`Math.sin(`*value*`)`
`Math.cos(`*value*`)`
`Math.tan(`*value*`)`
	sine/cosine/tangent of an angle in radians

`Math.toDegrees(`*value*`)` `Math.toRadians(`*value*`)`	convert degrees to radians and back

Constant	Description
`Math.E`	2.7182818...
`Math.PI`	3.1415926...

```
import java.util.Scanner;
public class Quadratic
{
   public static void main (String[] args)
   {
      int a, b, c;  // ax^2 + bx + c
      double discriminant, root1, root2;
      Scanner scan = new Scanner (System.in);

      System.out.print ("Enter the coefficient of x squared: ");
      a = scan.nextInt();
      System.out.print ("Enter the coefficient of x: ");
      b = scan.nextInt();
      System.out.print ("Enter the constant: ");
      c = scan.nextInt();

      // Use quadratic formula to compute the roots.

      discriminant = Math.pow(b, 2) - (4 * a * c);
      root1 = ((-1 * b) + Math.sqrt(discriminant)) / (2 * a);
      root2 = ((-1 * b) - Math.sqrt(discriminant)) / (2 * a);

      System.out.println ("Root #1: " + root1);
      System.out.println ("Root #2: " + root2);
   }
}
```

Trigonometric Methods

- `sin(double a)`
- `cos(double a)`
- `tan(double a)`
- `acos(double a)`
- `asin(double a)`
- `atan(double a)`

Examples:

```
Math.sin(0) returns 0.0
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/2) returns 0
Math.cos(Math.PI/6) returns 0.866
```

Exponent Methods

- **`exp(double a)`**
  Returns `e` raised to the power of `a`.
- **`log(double a)`**
  Returns the natural logarithm of `a`.
- **`log10(double a)`**
  Returns the 10-based logarithm of `a`.
- **`pow(double a, double b)`**
  Returns `a` raised to the power of `b`.
- **`sqrt(double a)`**
  Returns the square root of `a`.

Examples:

```
Math.exp(1) returns 2.71
Math.log(2.71) returns 1.0
Math.pow(2, 3) returns 8.0
Math.pow(3, 2) returns 9.0
Math.pow(3.5, 2.5) returns
   22.91765
Math.sqrt(4) returns 2.0
Math.sqrt(10.5) returns 3.24
```

Min(), max(), and abs()

- **`max(a,b)`** and **`min(a,b)`**

  Returns the maximum or minimum of two parameters.

- **`abs(a)`**

  Returns the absolute value of the parameter.

Examples:

```
Math.max(2,3) returns 3
Math.max(2.5,3) returns 3.0
Math.min(2.5,3.6) returns 2.5
Math.abs(-2) returns 2
Math.abs(-2.1) returns 2.1
```

Method `random()`

Generates a random double value greater than or equal to 0.0 and less than 1.0 (`0.0 <= Math.random() < 1.0`)

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.

NEXT SESSION: ARRAY