

6. Mathematical Library Methods

Introduction

1. Define Library Methods
Pre-defined methods developed by Java developers are **library methods** or **built-in methods**.
2. Define **Math** class
Math is a **pre-defined class** that consist some **pre-defined mathematical methods** to do certain complicated mathematical computations such as finding square root etc.
E.g.: **Math.sqrt(n)**;
3. In which **package** the Math class resides?
The Math class resides in **java.lang** package.
4. Is it necessary to **import** the **java.lang** package? Why?
It is **not necessary to import** the **java.lang** package into our program to use its methods since it is a **default package** (i.e., automatically it will be imported to our program).

Practice Questions

1. **Answer the followings:**
 - 1) Name the **pre-defined class** that resides mathematical library methods
Answer:
Math
 - 2) Name the **package** that resides Math class
Answer:
java.lang
 - 3) Name the **default package** of Java / automatically connected to a program of user
Answer:
java.lang

Methods and Use

Following 5 methods' output values are **double**

1. pow()

Use: To find the value of a^b

E.g.: Math.pow(2,3)

Output: 8.0

1. Write output for the followings:

- 1) int a=3,b=4; System.out.println(Math.pow(a,b));
Answer:
81.0
- 2) int a=2,b = -2; System.out.println(Math.pow(a,b));
Answer:
0.25 (2^{-2} is $1/2^2$, i.e., 2^2 is 4.0. $1/4.0$ is 0.25)
- 3) int a = -2, b=2; double n= Math.pow(a,b); System.out.println(n);
Answer:
4.0
- 4) int a=2,b=0; double n= Math.pow(a,b); System.out.println(n);
Answer:
1.0
- 5) int a=16; double b=1/2.0; System.out.println(Math.pow(a,b));
Answer:
4.0 ($1/2.0$ is 0.5. $16^{1/2.0}$ $16^{0.5}$ is 4.0 Square root)
- 6) int a=16, b=1/2; System.out.println(Math.pow(a,b));
Answer:
1.0 ($1/2$ is 0. 16^0 is 1.0)
- 7) int a=27; double b=1/3.0; double n= Math.pow(a,b); System.out.println(n);
Answer:
3.0 ($1/3.0$ is 0.3333333333. $27^{0.3333333333}$ is 3.0 Cube root)
- 8) **Write a statement:** To find fifth root of a number **n** and store it into **x**.
Answer:
double x=Math.pow(n, 1/5.0);
- 9) **Correct errors if any:** int x=Math.pow(4,3);
Answer:
double x=Math.pow(4,3);

2. sqrt()

Use: To find square root of a number

E.g.: Math.sqrt(36)

Output: 6.0

Note the output a of negative value:

E.g.: Math.sqrt(-36)

Output: NaN (It means **Not a Number**)

1. Write output for the followings:

1) int a=81; System.out.println(Math.sqrt(a));

Answer:

9.0

2) int a=-25; System.out.println(Math.sqrt(a));

Answer:

NaN

3) int a=2, b=4; System.out.println(Math.sqrt(Math.pow(a,b)));

Answer:

4.0

4) int a=4,b=3; System.out.println(Math.pow(b,Math.sqrt(a)));

Answer:

9.0

3. cbrt()

Use: To find cube root of a number

E.g.: Math.cbrt(27)

Output: 3.0

1. Write output for the followings:

1) int a=64; System.out.println(Math.cbrt(a));

Answer:

4.0

2) int a=-8; System.out.println(Math.cbrt(a));

Answer:

-2.0

3) int a=4, b=3; System.out.println(Math.cbrt(Math.pow(a,b)));

Answer:

4.0

4) int a=8,b=3; System.out.println(Math.pow(b,Math.cbrt(a)));

Answer:

9.0

4. floor()

Use: To round down a real value

E.g.: Math.floor(2.7)

Output: 2.0

E.g.: Math.floor(-2.7)

Output: -3.0 (Note that -3.0 is less than -2.7)

1. Write output:

1) double a=7.8; System.out.println(Math.floor(a));

Answer:

7.0

2) double a=-6.75; System.out.println(Math.floor(a));

Answer:

-7.0

5. ceil()

Use: To round up a real value

E.g.: Math.ceil(2.3)

Output: 3.0

E.g.: Math.ceil(-2.3)

Output: **-2.0** (Note that -2.0 is greater than -2.3)

1. Write output:

1) double a=7.2; System.out.println(Math.ceil(a));

Answer:

8.0

2) double a=-6.25; System.out.println(Math.ceil(a));

Answer:

-6.0

Following method's output value is long**6. round()**

Use: To **round off** a real value

E.g.: Math.round(**2.5**)

Output: **3** (**From .5** rounds up)

E.g.: Math.round(**2.57**)

Output: **3**

E.g.: Math.round(**2.49**)

Output: **2** (**Below .5** rounds down)

The return type of **round()** method is **long**. **int a=Math.round(2.5);** is **invalid**

1. Correct errors if any and write output:

1) int x=Math.round(7.4);

Correction:

long x=Math.round(7.4);

Output:

7

2) double a=Math.round(2.5);

Answer:

No error. (*The long can be stored into double*)

Output:

3.0

3) float a=Math.round(2.5);

Answer:

No error. (*The long can be stored into float*)

Output:

3.0

2. Write output:

1) double a=7.49; System.out.println(Math.round(a));

Answer:

7

2) double a=7.49; double b=Math.round(a); System.out.println(b);

Answer:

7.0

3) double a=7.5; System.out.println(Math.round(a));

Answer:

8

4) double a=7.58; double b= Math.round(a); System.out.println(b);

Answer:

8.0

5) double a=-6.25; System.out.println(Math.round(a));

Answer:

-6 (-6.25 is bigger than -6.5. So -6.25 rounds up. The -6 is bigger than -6.25)

6) double a=-6.25; double b=Math.round(a); System.out.println(b);

Answer:

-6.0

7) double a=-6.75; System.out.println(Math.round(a));

Answer:

-7 (-6.75 is lesser than -6.5. So -6.75 rounds down. -7 is lesser than -6.75)

8) double a=-6.75; double b= Math.round(a); System.out.println(b);

Answer:

-7.0

Following method's output value is double**7. random()**

Use: To find a random number (any number) between 0 and 1. No argument in ().

E.g.: Math.random() Output: **0.59365897451236**

1. Write any possible output and write what is it:

- 1) System.out.println(Math.random());

Answer:

0.295264620490289 (any number between 0 and 1)

- 2) System.out.println(1+Math.random());

Answer:

1.295264620490289 (any number between 1 and 2)

- 3) System.out.println((int)(Math.random()*10));

Answer:

5 (any number from 0 to 9)

Explanation:

Case 1: Math.random(); (0.57654364534267 * 10) is 5.7654364534267. Its (int) is 5

Case 2: Math.random(); (0.00654364534267 * 10) is 0.0654364534267. Its (int) is 0

Case 3: Math.random(); (0.95654364534267 * 10) is 9.5654364534267. Its (int) is 9

Following 3 methods' outputs are int or double according to their arguments**8. abs()**

Use: To find absolute (positive value)

E.g.: Math.abs(-5) Output: 5

E.g.: Math.abs(5) Output: 5

E.g.: Math.abs(-2.0) Output: 2.0

E.g.: Math.abs(2.6) Output: 2.6

1. Write output:

- 1) Math.abs(-6.8)

- 2) Math.abs(-59)

Answer:

6.8

Answer:

59

- 3) Math.abs(7)

- 4) Math.abs(4.6)

Answer:

7

Answer:

4.6

9. max()

Use: To find maximum of 2 values. If both arguments are int then returns int type

E.g.: Math.max(4,6) Output: 6

If one of the arguments or both are double then returns double

E.g.: Math.max(4.0,6) Output: 6.0

1. Write output:

- 1) Math.max(Math.max(5,7),8)

- 2) Math.max(Math.max(5.0,7),8)

Answer:

8 (maximum among 3 numbers)

Answer:

8.0

10. min()

Use: To find the minimum of 2 values. If both arguments are int then returns int type

E.g.: Math.min(4,6) Output: 4

If one of the arguments or both of them are double then returns double type.

E.g.: Math.min(4,6.0) Output: 4.0

1. Write output:

- 1) Math.min(Math.min(5,7),4)

- 2) Math.min(Math.min(5,7),8.0)

Answer:

4 (minimum among 3 numbers)

Answer:

5.0

Expressions Using Math Methods**1. Write corresponding Java expressions**

- 1) $\sqrt{a^2 + b^n}$

Math.sqrt(a*a+Math.pow(b,n))

- 2) $\frac{x+y}{mn}$

(x+y) / (m*n)

- 3) $\frac{p+q}{(r+s)^n}$

(p + q) / Math.pow(r+s,n)

- 4) $\frac{p-q}{r+s^n}$

(p-q) / (r+Math.pow(s,n))

- 5) $|x - y|$

Math.abs(x - y)