Mathematical Library Methods

1. 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);**

2. In which **package** the Math class resides?

   The Math class resides in **java.lang** package.

3. 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 running (user's) program
   Answer:
   java.lang

Methods and Use

Following 7 methods (except round()) outputs double data type value.

No.	Methods	Use	Example	Output
1	**pow()**	To find the value x^y	Math.pow(2,3)	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 4.0. 1/4.0 is 0.25)
3) int a=2,b=0; System.out.println(Math.pow(a,b));
   Answer:
   1.0
4) int a=16; double b=1/2.0; System.out.println(Math.pow(a,b));
   Answer:
   4.0 (1/2.0 is 0.5. $16^{0.5}$ is 4.0 Square root)
5) 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)
6) int a=27; double b=1/3.0; System.out.println(Math.pow(a,b));
   Answer:
   3.0 (1/3.0 is 0.33333333333. $27^{0.333333333}$ is 3.0. Cube root)

2. Correct errors if any:

1) int x=Math.pow(4,3);
   Answer:
   double x=Math.pow(4,3);
2) double x=Math.pow(4F,-3.0);
   Answer:
   No error.

2. **sqrt()** To find the square root of x Math.sqrt(36) 6.0

Note: The output of negative value: Math.sqrt(**-36**) is **NaN** (It means **N**ot **a N**umber*)*

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()** To find cube root of x Math.cbrt(27) 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

4. **floor()** To **round down** a real value Math.floor(2.7) 2.0
Math.floor(-2.7) **-3.0**
Note that -3.0 is less than -2.0

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()** To **round up** a real value Math.ceil(2.3) 3.0
Math.ceil(-2.3) **-2.0**
Note that -2.0 is greater than -3.0

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

6. **round()** To **round off** a real value. Output **long** type value
(**From .5** rounds up) Math.round(**2.5**) **3**
(**Below .5** rounds down) Math.round(**2.49**) **2**

Note: The return type of **round()** method is **long** data type.
int a=Math.round(2.5); is **invalid**. The method cannot be initialized to **int** variable. It requires a **long** variable. **long a=Math.round(2.5);** is valid. Output is **3**
double a=Math.round(2.5); and **float a=Math.round(2.5);** are valid because **long** value can be stored in both **double** and **float**. Output is **3.0**

Math.round(**-2.5**) is **-2** because the **-2** is **greater** than **-2.5**
Math.round(**-2.49**) is **-2** because the **-2.49** is **greater** than **-2.5**
Math.round(**-2.51)** is **-3** because the **-2.51** is **lesser** than **-2.5**

1) Write output:
double a=7.49; System.out.println(Math.round(a));
Answer:
7
2) Write output:
double a=7.49; double b= Math.round(a); System.out.println(b);
Answer:
7.0
3) Write output:
double a=7.5; System.out.println(Math.round(a));
Answer:
8
4) Write output:
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)
5) Write output:
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)

7. **random()** To find a random number (any number) between **0.0** and **1.0**
Math.random() **0.35465897451236**

1) Write the possible output values:
System.out.println(**Math.round()+1**);
Answer:
between 1.0 and 2.0
Explanation:
If the output value of Math.random() is **0.0143278453** then **+1** prints **1. 0143278453**
If the output value of Math.random() is **0.9143278453** then **+1** prints **1. 9143278453**
Note: The **Math.round()+1** is to return random numbers beteen **1.0** and **2.0**
2) Write the possible output values:
System.out.println(**(int)(Math.round()*10)**);
Answer:
from 0 to 9
Explanation:
If the output value of Math.random() is **0.0143278453** then ***10** is **0.143278453** and when it is typecasted to **int** it is **0**
If the output value of Math.random() is **0.9143278453** then ***10** is **9.143278453** and when it is typecasted to **int** it is **9**
Note: The **(int)(Math.round()*10)** is to return random numbers beteen **0** and **9**

Following 3 methods outputs int or double value according to their arguments

8.	**abs()**	To find absolute (positive value)	Math.abs(-5)	5
			Math.abs(5)	5
			Math.abs(-2.0)	2.0
			Math.abs(2.6)	2.6
9.	**max()**	To find maximum of 2 values	Math.max(4,6)	6
			Math.max(**4.0**,6)	**6.0**

Note: If both arguments are **int** then returns **int** type.
If **one of the arguments or both of them** are **double** then returns **double** type.

10.	**min()**	To find the minimum of 2 values	Math.min(4,6)	4
			Math.min(4,**6.0**)	**4.0**

Note: If both the arguments are **int** then returns **int** type value.
If **one of the arguments or both of them** are **double** then returns **double** type.

1. Write output:
   System.out.println(Math.max(Math.max(5,7),8));
   Answer:
   8
   Note: The above expression is to find maximum among three numbers.
1. Write output:
   System.out.println(Math.min(Math.min(5,7),4));
   Answer:
   4
   Note: The above expression is to find minimum among three numbers.

Methods to Write Equivalent Java Expressions

Following functions returns double data type value

11.	log()	To find logarithm of x	Math.log(x)
12.	sin()	To find the **sine** of the angle x	Math.sin(x)
13.	cos()	To find the **cosine** of the angle x	Math.cos(x)
14.	tan()	To find the **tangent** of the angle x	Math.tan(x)
15.	asin()	To find the **angle** whose sine is y	Math.asin(y)
16.	acos()	To find **angle** whose cosine is y	Math.acos(y)
17.	atan()	To find **angle** whose tangent is y	Math.atan(y)
18.	exp()	To find the value of e raised to x	Math.exp(x)

Expressions Using Math Methods

1. Write corresponding Java expressions

1) $\sqrt{a^2 + b^2 + c^n}$	Math.sqrt(a*a + b*b + Math.pow(c,n))
2) $\sqrt{a^n + b^n}$	Math.sqrt(Math.pow(a,n)+Math.pow(b,n))
3) $\frac{x+y}{mn}$	(x+y) / (m*n)
4) $\frac{p+q}{(r+s)^n}$	(p + q) / Math.pow(r+s,n)
5) $\frac{p-q}{r+s^n}$	(p-q) / (r+Math.pow(s,n))
6) $2 - ye^{2y} + 4y$	2 – y * **Math.exp(2*y)** + 4 * y
7) 2 – **ye** + 4y	2 – **y * e** + 4 * y
8) log x * tan 2y – a	Math.log(x) * Math.tan(2*y) – a
9) **|**e – x**|**	Math.abs(e – x)
10) **|**e^x – x**|**	Math.abs(**Math.exp(x)** – x)