JAVA.LANG.MATH.SCALB METHOD

Description

The **java.lang.Math.scalb*floatf, intscaleFactor*** returns f x $2^{scaleFactor}$ rounded as if performed by a single correctly rounded floating-point multiply to a member of the double value set. See the Java Language Specification for a discussion of floating-point value sets. If the exponent of the result is between Float.MIN_EXPONENT and Float.MAX_EXPONENT, the answer is calculated exactly. If the exponent of the result would be larger than Float.MAX_EXPONENT, an infinity is returned. Note that if the result is subnormal, precision may be lost; that is, when scalbx, n is subnormal, scalb$scalb(x, n, -n)$ may not equal x. When the result is non-NaN, the result has the same sign as f.

- If the first argument is NaN, NaN is returned.
- If the first argument is infinite, then an infinity of the same sign is returned.
- If the first argument is zero, then a zero of the same sign is returned.

Declaration

Following is the declaration for **java.lang.Math.scalb** method

```
public static double scalb(float f, int scaleFactor)
```

Parameters

- **f** -- number to be scaled by a power of two.
- **scaleFactor** -- power of 2 used to scale d

Return Value

This method returns f x $2^{scaleFactor}$

Exception

- **NA**

Example

The following example shows the usage of lang.Math.scalb method.

```
package com.tutorialspoint;

import java.lang.*;

public class MathDemo {

   public static void main(String[] args) {

   // get a x to be raised
   float x = 50.14f;
   int y = 4;

   // calculate x multiplied by 2 raised in y
   System.out.println("Math.scalb(" + x + "," + y + ")=" + Math.scalb(x, y));
   }
}
```

Let us compile and run the above program, this will produce the following result:

```
Math.scalb(50.14f, 4)=802.24
```