NAME: Thejus Gowda PN
email: thejusgowdapn1059@gmail.com

POW method

It is used to calculate a number raise to the power of some other number. This function accepts two parameters and returns the value of first parameter raised to the second parameter . There are some special cases listed below .

- If the second parameter is positive or negative zero then the result will be 1.0
- If the second parameter is 1.0 then the result will be same as that of the first parameter
- If the second parameter is NaN then the result will also be NaN.
- The function java.lang.Math.pow() always returns a double datatype.

```
public class Math {
    // ...

    public static double pow(double base, double exponent) {
        // Handle special cases
        if (exponent == 0) {
            return 1.0; // Any number raised to the power of 0 is 1
        }
        if (base == 0) {
            return 0.0; // 0 raised to any power (except 0) is 0
        }

        // Calculate using the property a^b = e^(b * ln(a))
        double result = java.lang.StrictMath.exp(exponent *
java.lang.StrictMath.log(base));

        return result;
    }

    // ...
}
```

Random Method

The java.lang.Math.Random method returns a pseudorandom double type number greater then or equal to 0.0 and less than 1.0 When this method is first called, it creates a single new pseudorandom-number generator , exactly as if by the expression new java.util.Random

example :

```
// Java program to demonstrate working
// of java.lang.Math.random() method
import java.lang.Math;

// Driver Class
class Gfg1 {
        // driver code
        public static void main(String args[])
        {
                // Generate random number
                double rand = Math.random();

                // Output is different everytime this code is executed
                System.out.println("Random Number:" + rand);
        }
}
```

Random method code

```
public class Math {
   // ...

   private static long seed = System.currentTimeMillis(); // Initial seed

   public static double random() {
      seed = (seed * 0x5DEECE66DL + 0xBL) & ((1L << 48) - 1);
      return (double) (seed >>> 17) / (1L << 31);
   }

   // ...
}
```