Math.pow( )

Assignment: 9

Date: 30-08-2023

Name: Vyshnavi H S

E-mail id: vyshnavirao5@gmail.com

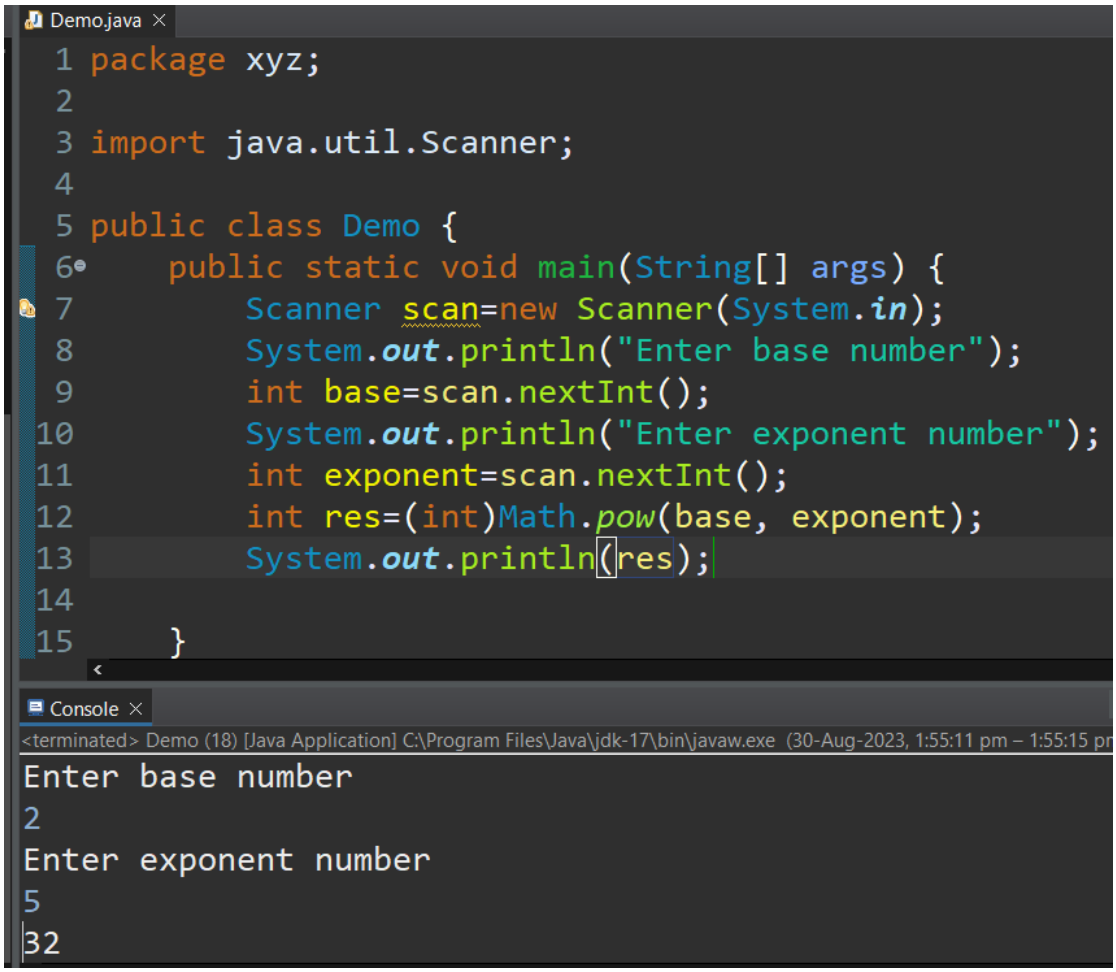
Math.pow( ):

The java.lang.Math.pow() is used to return the value of first argument raised to the power of the second argument. The return type of pow() method is double.

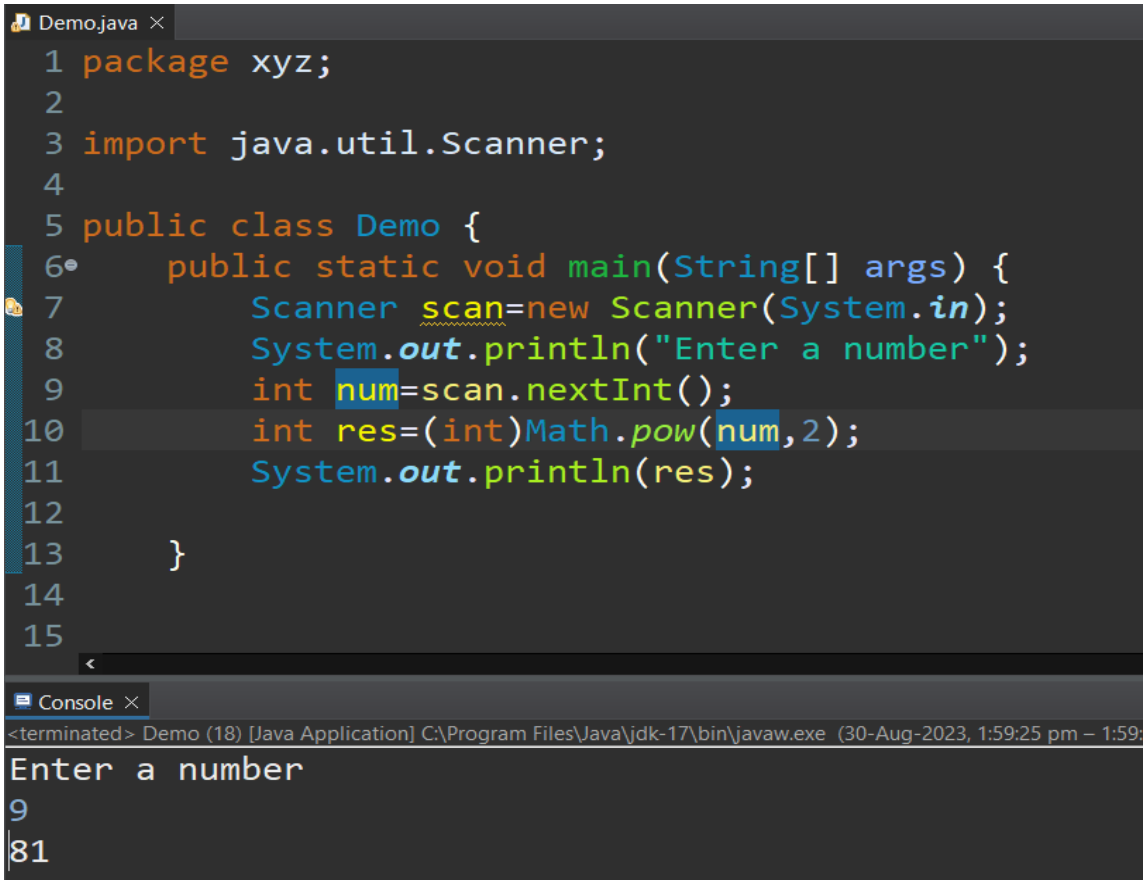
- public static double pow(double a, double b)
- a= base, b= exponent.

Uses:

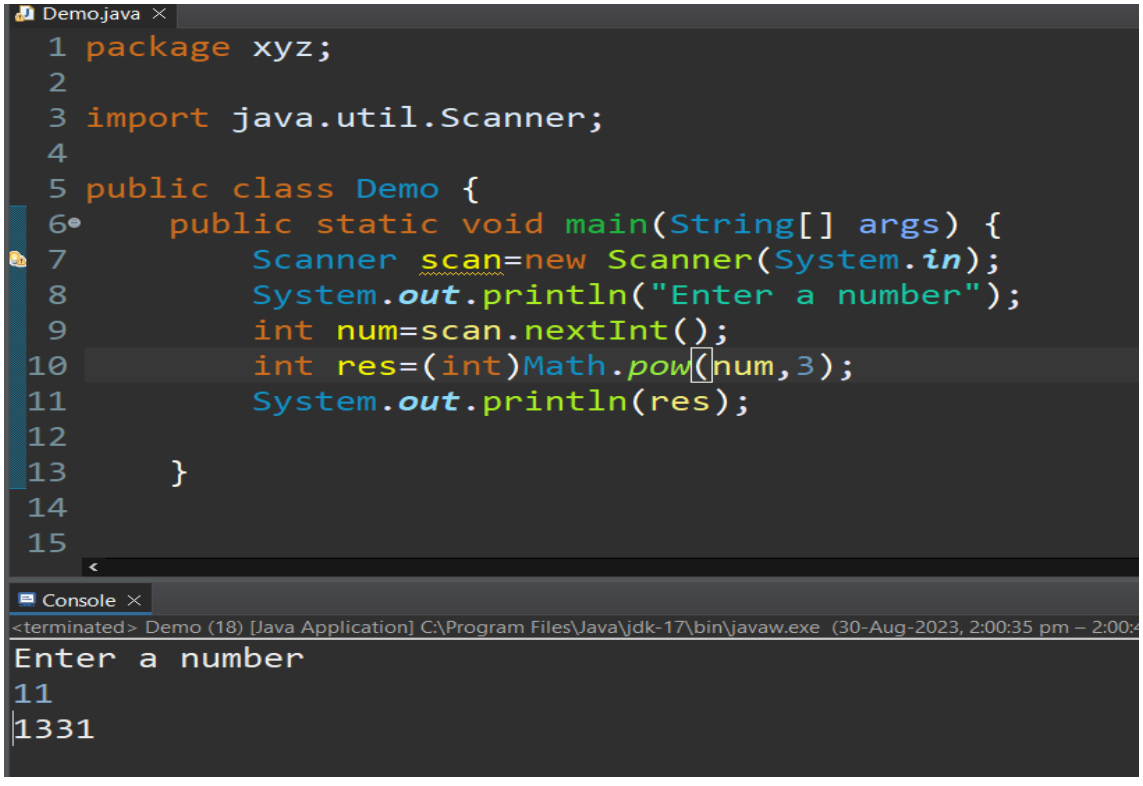
1. Exponential calculations.



2. To find square of a number



3. To find cube of a number



4. To find Square root of a number

Demo.java

```
1  package xyz;
2
3  import java.util.Scanner;
4
5  public class Demo {
6      public static void main(String[] args) {
7          Scanner scan=new Scanner(System.in);
8          System.out.println("Enter a number");
9          int num=scan.nextInt();
10         int res=(int)Math.pow(num,0.5);
11         System.out.println(res);
12
13     }
14
15
```

Console

```
<terminated> Demo (18) [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (30-Aug-2023, 2:32:06 pm – 2:32
Enter a number
64
8
```

Demo.java

```
3  import java.util.Scanner;
4
5  public class Demo {
6      public static void main(String[] args) {
7          Scanner scan=new Scanner(System.in);
8          System.out.println("Enter a number");
9          int num=scan.nextInt();
10         double nan = Double.NaN;
11         //if the second parameter is NaN result will be NaN
12         double result1 =Math.pow(num,nan);
13         System.out.println(result1);
14
15         // if the second parameter is zero, result will be 1
16         int result2 =(int)Math.pow(num, 0);
17         System.out.println(result2);
18
19         // if the second paramater is one, result will
20         //be same as first parameter
21        int result3 = (int)Math.pow(num, 1);
22         System.out.println(result3);
23         }}
```

Console

```
<terminated> Demo (18) [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (30-Aug-2023, 2:46:29 pm – 2:4
Enter a number
12345
NaN
1
12345
```

- If the second parameter is 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.

Math.random( )

Math.random( )

The java.lang.Math.random() is used to return a pseudorandom double type number greater than or equal to 0.0 and less than 1.0. The default random number always generated between 0 and 1.

Demo.java

```
3  import java.util.Scanner;
4
5  public class Demo {
6      public static void main(String[] args) {
7          // Generate random number between 0.0 to 1.0
8          double a1 = Math.random();
9          double b1 = Math.random();
10         // Output is different every time this code is executed
11         System.out.println("Random number between 0.0 to 1.0 = "+a1);
12         System.out.println("Random number between 0.0 to 1.0 = "+b1);
13     // Generate random number between 0 to 20
14         double a2 = Math.random() * 20;
15         double b2 = Math.random() * 20;
16         System.out.println("random number between 0 to 20 = "+a2);
17         System.out.println("random number between 0 to 20 = "+b2);
18     // Generate random number between 5 to 30
19         int a3 =(int)(5 + (Math.random() * 30));
20         int b3 = (int)(5 +  (Math.random() * 30));
21         System.out.println("random number between 5 to 30 = "+a3);
22         System.out.println("random number between 5 to 30 = "+b3);
23         }}
24
```

Console

```
<terminated> Demo (18) [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (30-Aug-2023, 3:42:36 pm – 3:42:3
Random number between 0.0 to 1.0 = 0.6782907269968524
Random number between 0.0 to 1.0 = 0.1549593030478419
random number between 0 to 20 = 6.116825151376986
random number between 0 to 20 = 14.2752152356741
random number between 5 to 30 = 11
random number between 5 to 30 = 14
```

- If we want a specific range of values, you have to multiply the returned value with the magnitude of the range. For example, if you want to get the random number between 0 to 20, the resultant address has to be multiplied by 20 to get the desired result.