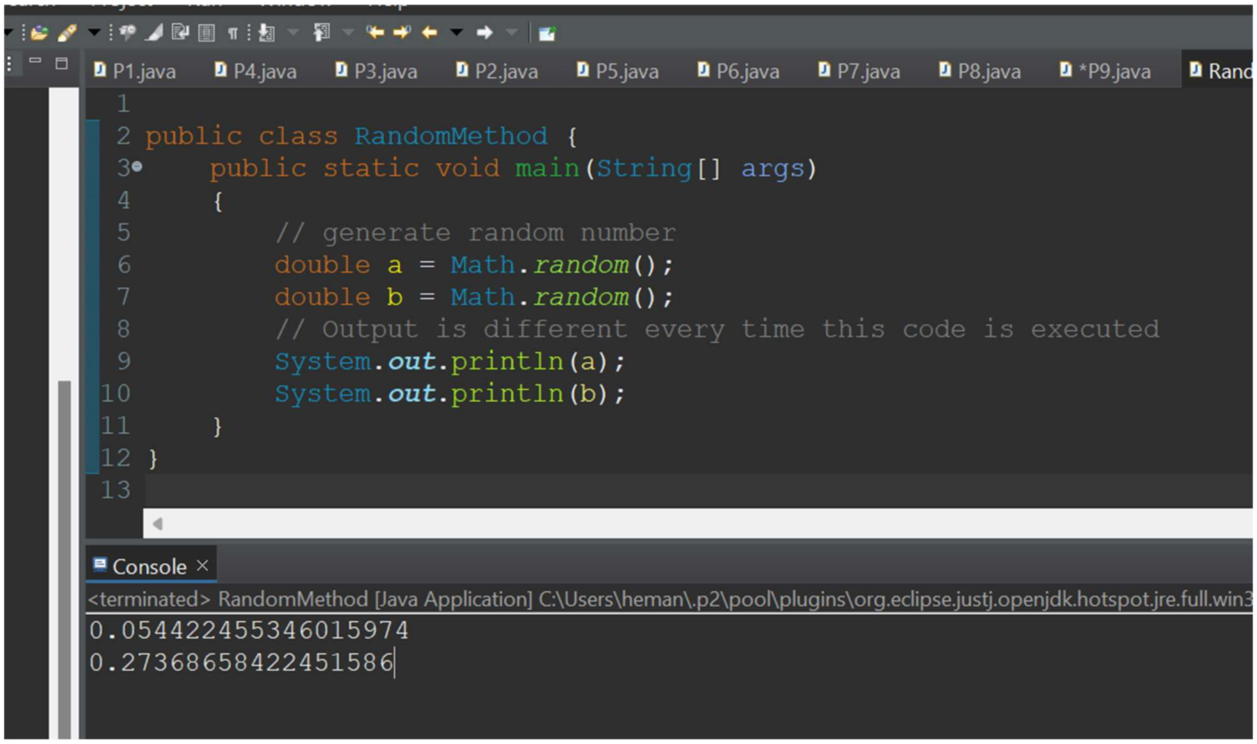
Java Math random() Method

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.

Example:-



```java

public class RandomMethod {
    public static void main(String[] args)
    {
        // generate random number
        double a = Math.random();
        double b = Math.random();
        // Output is different every time this code is executed
        System.out.println(a);
        System.out.println(b);
    }
}

```

```
<terminated> RandomMethod [Java Application] C:\Users\heman\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win3
0.054422455346015974
0.27368658422451586
```

Application of random numbers

There are times when you need to generate a random number in programming. For example, say that you are operating a cruise line, as a booking reference, you may want to add a random number to a customer's order.

The Math.random() method in Java has many applications in various fields and industries. Some of the most common use cases for Math.random() include:

i.) Gaming: The Math.random() method is often used to generate random numbers for game development, such as random events, dice rolls, or card shuffling.

ii.) Statistical Analysis: Random numbers generated by Math.random() can be used in statistical analysis, such as Monte Carlo simulations, to model and predict outcomes.

iii.) Cryptography: In cryptography, random numbers generated by Math.random() can be used as keys or seeds to encrypt or decrypt sensitive information.

iv.) Testing: Math.random() can be used to generate random test data for software development, allowing developers to test their applications in a variety of scenarios.

v.) Artificial Intelligence: Math.random() can also be used in artificial intelligence and machine learning applications, such as genetic algorithms and neural networks, to generate random inputs for training and testing.

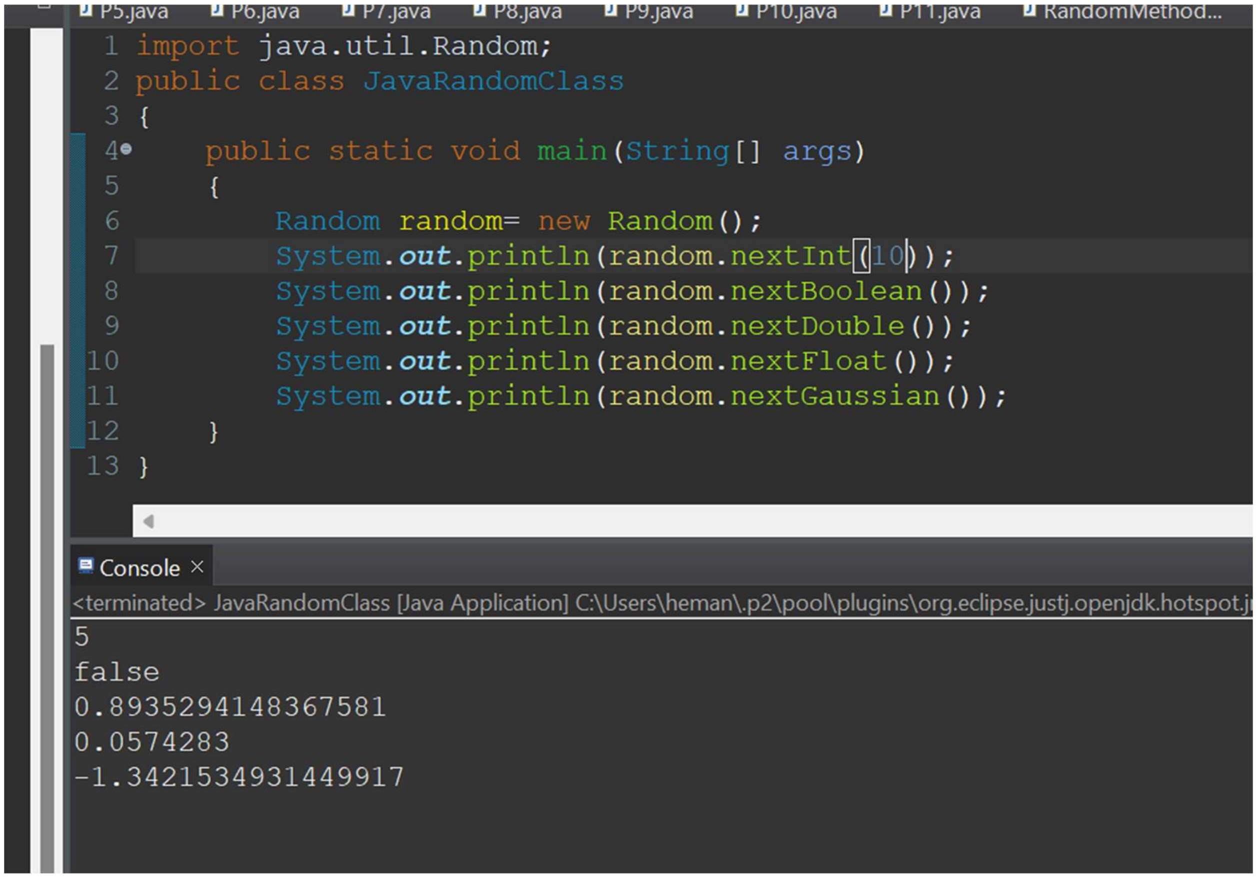
Advantages of using Math.random() in Java

1. **Simplicity:** Math.random() is a simple and easy to use method for generating random numbers.
2. **Flexibility:** It can be used to generate random numbers of various types, such as integers or decimals, within a specified range.
3. **Widely used:** It is widely used in various applications, such as gaming, simulation, and statistical analysis.

Disadvantages of using Math.random() in Java

1. **Predictability:** The sequence of random numbers generated by Math.random() can be predictable if not used correctly.
2. **Limited range:** Math.random() only generates random numbers between 0 and 1, and the range must be scaled to meet the needs of a specific application.
3. **Non-uniform distribution:** Math.random() generates random numbers with a non-uniform distribution, which can affect the accuracy of certain applications.
4. **Seed dependence:** The random numbers generated by Math.random() are dependent on the seed value, and if the same seed is used, the same sequence of random numbers will be generated.

Java random Class



P5.java P6.java P7.java P8.java P9.java P10.java P11.java RandomMethod...

```java
import java.util.Random;
public class JavaRandomClass
{
    public static void main(String[] args)
    {
        Random random= new Random();
        System.out.println(random.nextInt(10));
        System.out.println(random.nextBoolean());
        System.out.println(random.nextDouble());
        System.out.println(random.nextFloat());
        System.out.println(random.nextGaussian());
    }
}
```

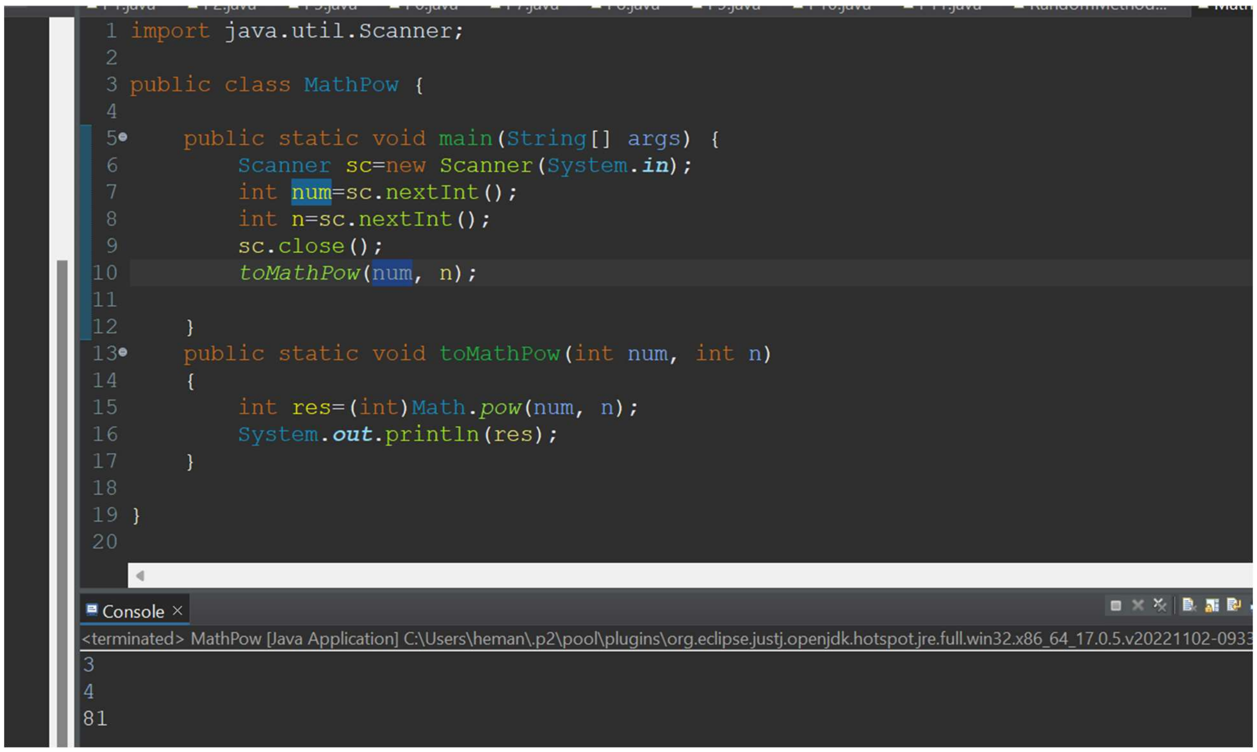
Console

```
<terminated> JavaRandomClass [Java Application] C:\Users\heman\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.j
5
false
0.8935294148367581
0.0574283
-1.3421534931449917
```

Java Math.pow() Method

This method returns the value of a^{b}

- If the second argument is positive or negative **Zero**, this method will return **1.0**.
- If the second argument is not a number **(NaN)**, this method will return **NaN**.
- If the second argument is **1**, this method will return the result same as the **first argument**.



```java
import java.util.Scanner;

public class MathPow {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int num=sc.nextInt();
        int n=sc.nextInt();
        sc.close();
        toMathPow(num, n);

    }
    public static void toMathPow(int num, int n)
    {
        int res=(int)Math.pow(num, n);
        System.out.println(res);
    }

}
```

```
<terminated> MathPow [Java Application] C:\Users\heman\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933
3
4
81
```