

## **CSE1007 Java Programming. Fall Semester. Class Activity. 15-October-2021**

Sharadindu Adhikari  
19BCE2105

### **Questions:**

**Q1. How do you calculate the total execution time of the java program?**

1. `currentTimeMillis()`
2. `nanoTime()`
3. `now()`

**Q2. How do you generate the random numbers in java?**

1. `random()` Method
2. Random Class
3. `ThreadLocalRandom` Class

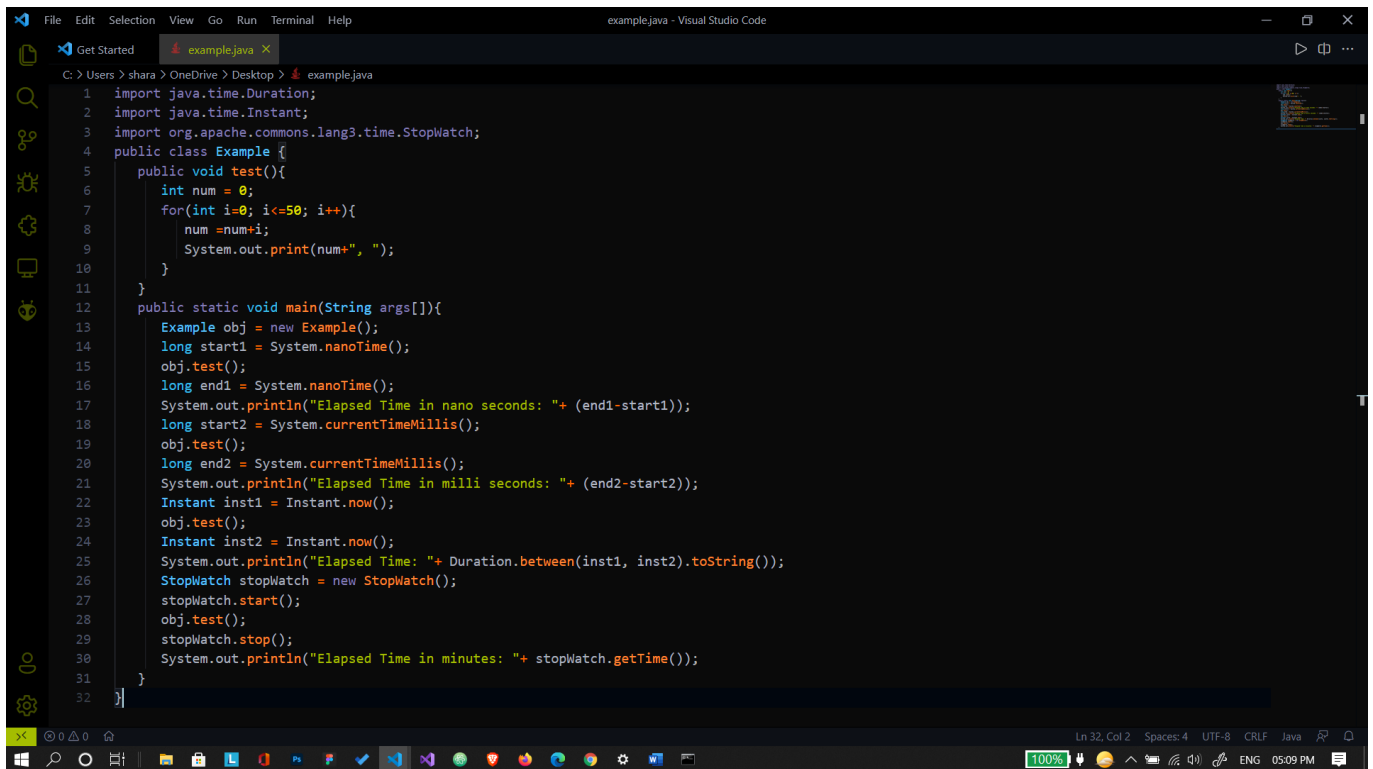
### **Solutions:**

**Q1. Calculating total execution time of the Java program using the given methods.**

In general, the elapsed time is the time from the starting point to ending point of an event. Following are the given ways to find elapsed time in Java:

- The `currentTimeMillis()` method returns the current time in milliseconds. To find the elapsed time for a method you can get the difference between time values before and after the execution of the desired method.
- The `nanoTime()` method returns the current time in nano seconds. To find the elapsed time for a method you can get the difference between time values before and after the execution of the desired method.
- The `now()` method of the `Instant` class returns the current time and the `Duration.between()` methods returns the difference between the given two time values to get the elapsed time retrieve the time values before and after the execution of the desired method and retrieve the duration using the `Duration.between()` method.

Following example demonstrates how to find the execution time of a method using the given methods:



```
File Edit Selection View Go Run Terminal Help
example.java - Visual Studio Code

C:\Users\shara> OneDrive\ Desktop > example.java
1 import java.time.Duration;
2 import java.time.Instant;
3 import org.apache.commons.lang3.time.StopWatch;
4 public class Example {
5     public void test(){
6         int num = 0;
7         for(int i=0; i<=50; i++){
8             num =num+i;
9             System.out.print(num+" ", "");
10        }
11    }
12    public static void main(String args[]){
13        Example obj = new Example();
14        long start1 = System.nanoTime();
15        obj.test();
16        long end1 = System.nanoTime();
17        System.out.println("Elapsed Time in nano seconds: "+ (end1-start1));
18        long start2 = System.currentTimeMillis();
19        obj.test();
20        long end2 = System.currentTimeMillis();
21        System.out.println("Elapsed Time in milli seconds: "+ (end2-start2));
22        Instant inst1 = Instant.now();
23        obj.test();
24        Instant inst2 = Instant.now();
25        System.out.println("Elapsed Time: "+ Duration.between(inst1, inst2).toString());
26        StopWatch stopWatch = new StopWatch();
27        stopWatch.start();
28        obj.test();
29        stopWatch.stop();
30        System.out.println("Elapsed Time in minutes: "+ stopWatch.getTime());
31    }
32 }
```

Ln 32, Col 2 Spaces: 4 UTF-8 CRLF Java 100%

## Output:

```
0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136, 153,
171, 190, 210, 231, 253, 276, 300, 325, 351, 378, 406, 435, 465, 496,
528, 561, 595, 630, 666, 703, 741, 780, 820, 861, 903, 946, 990, 1035,
1081, 1128, 1176, 1225, 1275, Elapsed Time in nano seconds: 1882300
0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136, 153,
171, 190, 210, 231, 253, 276, 300, 325, 351, 378, 406, 435, 465, 496,
528, 561, 595, 630, 666, 703, 741, 780, 820, 861, 903, 946, 990, 1035,
1081, 1128, 1176, 1225, 1275, Elapsed Time in milli seconds: 1
0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136, 153,
171, 190, 210, 231, 253, 276, 300, 325, 351, 378, 406, 435, 465, 496,
528, 561, 595, 630, 666, 703, 741, 780, 820, 861, 903, 946, 990, 1035,
1081, 1128, 1176, 1225, 1275, Elapsed Time: PT0.001S
0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136, 153,
171, 190, 210, 231, 253, 276, 300, 325, 351, 378, 406, 435, 465, 496,
528, 561, 595, 630, 666, 703, 741, 780, 820, 861, 903, 946, 990, 1035,
1081, 1128, 1176, 1225, 1275, Elapsed Time in minutes: 1
```

**Q2. In Java, there are three main ways to generate random numbers using the method and classes.**

- Using the random() Method
- Using the Random Class
- Using the ThreadLocalRandom Class

## 1. Using the random() method

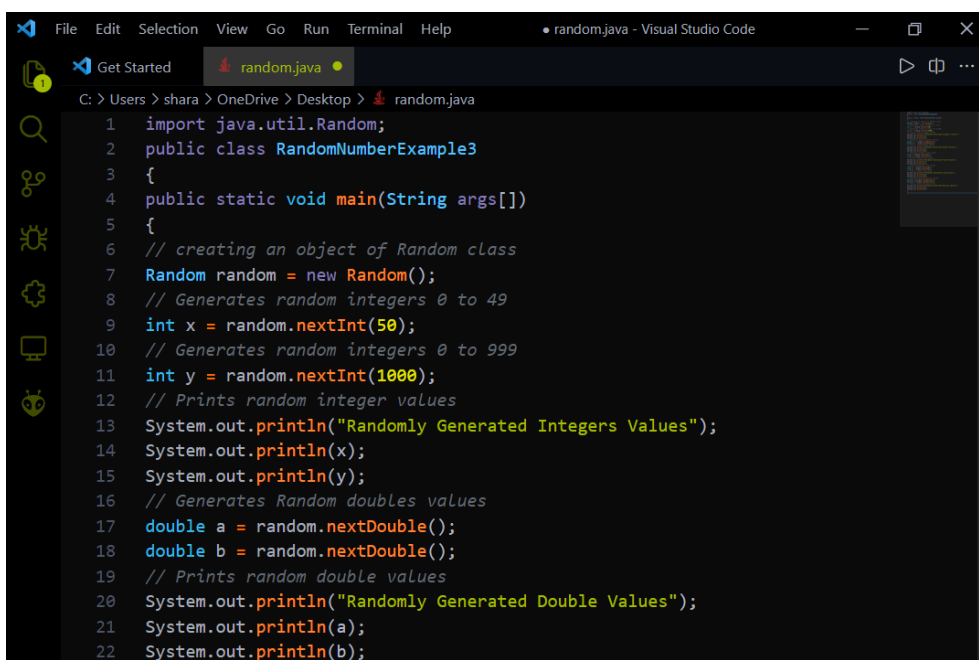


```
1 import java.lang.Math;
2 public class RandomNumberExample1
3 {
4     public static void main(String args[])
5     {
6         // Generating random numbers
7         System.out.println("1st Random Number: " + Math.random());
8         System.out.println("2nd Random Number: " + Math.random());
9         System.out.println("3rd Random Number: " + Math.random());
10        System.out.println("4th Random Number: " + Math.random());
11    }
12 }
```

Output:

```
1st Random Number: 0.17434160924512265
2nd Random Number: 0.4297410090709448
3rd Random Number: 0.4828656381344487
4th Random Number: 0.13267917059488898
```

## 2. Using the Random class



```
1 import java.util.Random;
2 public class RandomNumberExample3
3 {
4     public static void main(String args[])
5     {
6         // creating an object of Random class
7         Random random = new Random();
8         // Generates random integers 0 to 49
9         int x = random.nextInt(50);
10        // Generates random integers 0 to 999
11        int y = random.nextInt(1000);
12        // Prints random integer values
13        System.out.println("Randomly Generated Integers Values");
14        System.out.println(x);
15        System.out.println(y);
16        // Generates Random doubles values
17        double a = random.nextDouble();
18        double b = random.nextDouble();
19        // Prints random double values
20        System.out.println("Randomly Generated Double Values");
21        System.out.println(a);
22        System.out.println(b);
23    }
24 }
```

```
26 // Prints random float values
27 System.out.println("Randomly Generated Float Values");
28 System.out.println(f);
29 System.out.println(i);
30 // Generates Random Long values
31 long p = random.nextLong();
32 long q = random.nextLong();
33 // Prints random Long values
34 System.out.println("Randomly Generated Long Values");
35 System.out.println(p);
36 System.out.println(q);
37 // Generates Random boolean values
38 boolean m=random.nextBoolean();
39 boolean n=random.nextBoolean();
40 // Prints random boolean values
41 System.out.println("Randomly Generated Boolean Values");
42 System.out.println(m);
43 System.out.println(n);
44 }
45 }
```

### Output:

Randomly Generated Integers Values

23

767

Randomly Generated Double Values

0.37823814494212016

0.998058172671956

Randomly Generated Float Values

0.87804186

0.93880254

Randomly Generated Long Values

-4974823544291679198

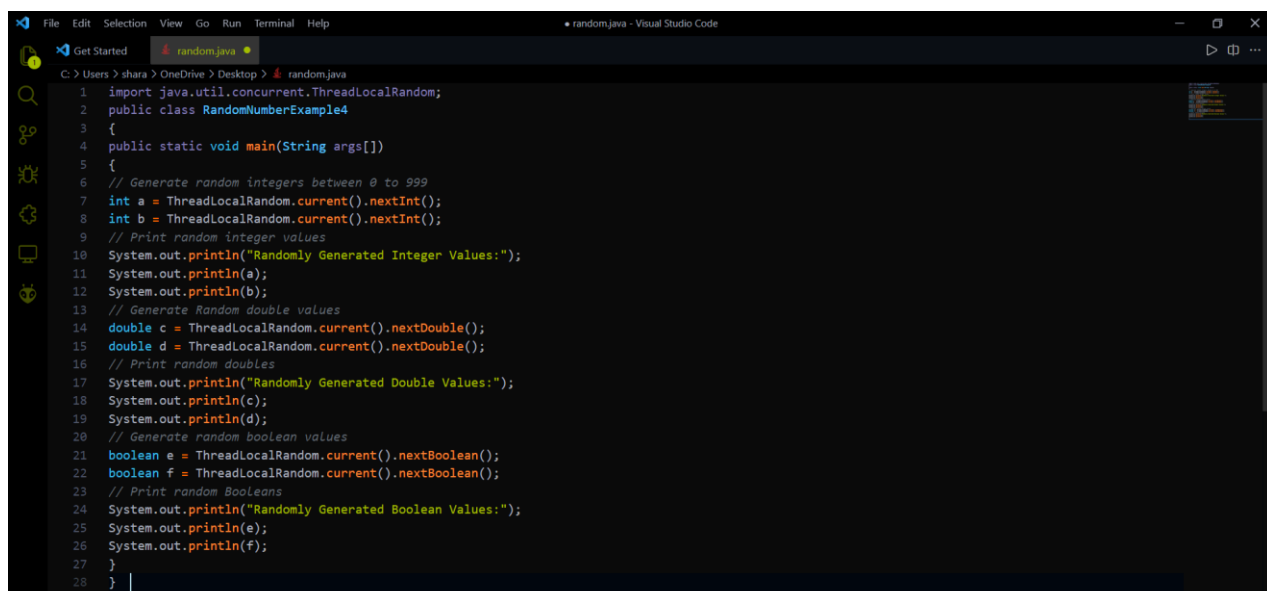
3650240138416076693

Randomly Generated Boolean Values

false

true

### 3. Using the ThreadLocalRandom Class



```
1 import java.util.concurrent.ThreadLocalRandom;
2 public class RandomNumberExample4
3 {
4     public static void main(String args[])
5     {
6         // Generate random integers between 0 to 999
7         int a = ThreadLocalRandom.current().nextInt();
8         int b = ThreadLocalRandom.current().nextInt();
9         // Print random integer values
10        System.out.println("Randomly Generated Integer Values:");
11        System.out.println(a);
12        System.out.println(b);
13        // Generate Random double values
14        double c = ThreadLocalRandom.current().nextDouble();
15        double d = ThreadLocalRandom.current().nextDouble();
16        // Print random doubles
17        System.out.println("Randomly Generated Double Values:");
18        System.out.println(c);
19        System.out.println(d);
20        // Generate random boolean values
21        boolean e = ThreadLocalRandom.current().nextBoolean();
22        boolean f = ThreadLocalRandom.current().nextBoolean();
23        // Print random Booleans
24        System.out.println("Randomly Generated Boolean Values:");
25        System.out.println(e);
26        System.out.println(f);
27    }
28 }
```

**Output 1:**

```
Randomly Generated Integer Values:  
348534891  
-1887936727  
Randomly Generated Double Values:  
0.15644440033119833  
0.5242730752133399  
Randomly Generated Boolean Values:  
true  
true
```

**Output 2:**

```
Randomly Generated Integer Values:  
402755574  
295398333  
Randomly Generated Double Values:  
0.4856461791062565  
0.5148677091077654  
Randomly Generated Boolean Values:  
false  
true
```

---