

# Thread.sleep() Method in Java With Examples

Difficulty Level : Basic Last Updated : 08 Dec, 2021

Thread Class is a class that is basically a thread of execution of the programs. It is present in Java.lang.package. Thread class contains the **Sleep()** method. There are two overloaded methods of Sleep() method present in Thread Class, one is with one argument and another one is with two arguments. The sleep() method is used to stop the execution of the current thread(whichever might be executing in the system) for a specific duration of the time and after that time duration gets over, the thread which is executing earlier starts to execute again.

## Important Point Regarding Thread.sleep() Method:

- Method Whenever Thread.sleep() functions to execute, it always pauses the current thread execution.
- If any other thread interrupts when the thread is sleeping, then InterruptedException will be thrown.
- If the system is busy, then the actual time the thread will sleep will be more as compared to that passed while calling the sleep method and if the system has less load, then the actual sleep time of the thread will be close to that passed while calling sleep() method.

## Syntax Of Sleep() Method

- public static void sleep(long millis)throws InterruptedException
- public static void sleep(long millis)throws IllegalArgumentException
- public static void sleep(long millis, int nanos)throws InterruptedException
- public static void sleep(long millis, int nanos)throws IllegalArgumentException

## Parameters Passed In Thread.Sleep() Method

- millis: Duration of time in milliseconds for which thread will sleep
- nanos: This is the additional time in nanoseconds for which we want the thread to sleep. It ranges from 0 to 999999.

**Return Type Of Sleep() Method:** It does not return any value, the ie the return type of the sleep function is void.

The sleep method with one parameter is a native method, ie the implementation of this method is done in another programming language and the other method with two parameters is not a native method, ie its implementation is done in Java. Both the sleep methods are static, ie we can access

them using the Thread class. Both the methods throw a checked Exception ie we can handle the exception either by using the throws keyword or by using try and catch block.

We can use Thread.Sleep() method for any thread, ie we can it with the main thread or any other thread that we make programmatically.

- **Using Thread.Sleep() Method For Main Thread**

---

```
// Java Program for sleeping the main thread.

import java.io.*;
import java.lang.Thread;

class GFG {
    public static void main(String[] args)
    {
        // we can also use throws keyword followed by
        // exception name for throwing the exception

        try {
            for (int i = 0; i < 5; i++) {

                // it will sleep the main thread for 1 sec
                // ,each time the for loop runs
                Thread.sleep(1000);

                // printing the value of the variable
                System.out.println(i);
            }
        } catch (Exception e) {

            // catching the exception
            System.out.println(e);
        }
    }
}
```

## Output

```
0
1
```

2  
3  
4

- **Using Thread.Sleep() Method For Custom Thread**

---

// Java Program for sleeping the custom thread.

```
import java.io.*;
import java.lang.Thread;

class GFG extends Thread {

    public void run()
    {
        // thread 0

        // we can also use throws keyword followed by
        // exception name for throwing the exception
        try {
            for (int i = 0; i < 5; i++) {

                // it will sleep the main thread for 1 sec
                // ,each time the for loop runs
                Thread.sleep(1000);

                // This Thread.sleep() method will sleep the
                // thread 0.
                // printing the value of the variable
                System.out.println(i);
            }
        }
        catch (Exception e) {

            // catching the exception
            System.out.println(e);
        }
    }

    public static void main(String[] args)
    {
        // main thread
        GFG obj = new GFG();
        obj.start();
    }
}
```

## Output

0  
1  
2  
3  
4

- **IllegalArgumentException When Sleep Time Is Negative**

---

```
// Java Program for showing how exception can occur if we  
// pass the negative timeout value.
```

```
import java.io.*;  
import java.lang.Thread;  
  
class GFG {  
    public static void main(String[] args)  
    {  
        // we can also use throws keyword followed by  
        // exception name for throwing the exception  
  
        try {  
            for (int i = 0; i < 5; i++) {  
  
                // this will throw the  
                // IllegalArgumentException  
                Thread.sleep(-100);  
  
                // printing the value of the variable  
                System.out.println(i);  
            }  
        }  
        catch (Exception e) {  
  
            // catching the exception  
            System.out.println(e);  
        }  
    }  
}
```

## Output

```
java.lang.IllegalArgumentException: timeout value is negative
```