

Java Multi-Threading Job Interview Questions And Answers



Interview Questions Answers

<https://interviewquestionsanswers.org/>

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Java Multi-Threading will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [Java Multi-Threading Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Java Multi-Threading category. To ensure quality, each submission is checked by our team, before it becomes live. This [Java Multi-Threading Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
<https://InterviewQuestionsAnswers.ORG/>
Support@InterviewQuestionsAnswers.ORG



Java Multi-Threading Interview Questions And Answers Guide.

Question - 1:

What is thread synchronization?

Ans:

In multithreading, the shared resources are allocated to one thread at a time. Multiple threads ensuring that the shared resources will be allocated to only one thread at a time is called thread synchronization. The synchronized keyword is used to define the critical block in which every Java object gets a lock to enter.

Syntax:

```
Synchronized(object) // object is reference of the object to be synchronized
{
// statement to be synchronized.
}
```

The synchronized keyword provides locking which maintains the mutual exclusive access of shared resource and race of objects for data.

[View All Answers](#)

Question - 2:

Explain the difference between preemptive scheduling and time slicing?

Ans:

In preemptive scheduling, highest priority tasks will execute until they enter waiting or dead states. It also executes until a higher priority task enters.

In time slicing, a task will execute for a fixed time slice and after that it will go into a ready state.

At that time the scheduler will find the executable task, according to the priority and various other tasks.

In preemptive scheduling, the running task will be preempted by the higher priority task.

In time slicing methods, a task executes until the specified period of time. Once the execution of that task is complete then the higher priority task will be executed if available.

[View All Answers](#)

Question - 3:

Explain the difference between yielding and sleeping?

Ans:

Sleep causes the currently executing thread to sleep until the specified time is completed. The thread will resume once the specified time period is over.

Sleep causes the currently executing thread to sleep and gives a chance to other threads to execute. The thread will join the ready queue.

Thread.sleep() will move the thread to "Wait" state.

Thread.yield() will move the thread to "Ready" state.

[View All Answers](#)

Question - 4:

What is the difference between notify and notifyAll methods?

Ans:

The notify will run only the first thread which has called wait() on the same object. i.e. the object which has called the wait() must be the same as the object that calls.

The notifyAll will run all threads which have called wait() on the same object.

While using notifyAll the highest priority thread will run first.

[View All Answers](#)

Question - 5:

Explain some ways in which a thread can enter the waiting state?

Ans:

A thread can enter the waiting state by the following ways:

We can invoke sleep() method of the thread.



An attempt to acquire the object's lock can put the thread in waiting mode.
We can also invoke wait() method of the thread.
A thread can also be entered in waiting state by invoking its suspend() method.

[View All Answers](#)

Question - 6:

Explain Thread?

Ans:

We can describe "thread" in two ways in Java.

1. An instance of class java.lang.Thread.
2. A thread of execution.

An instance of Thread is an object like all other objects in Java, it also has variables and methods and it lives and dies on the heap.

But a thread of execution is an individual light weight process that has its own call stack.

Even if you don't create any new threads in your program, threads are there running in background.

The main() method runs the main thread. So in main call stack the main() method would be the first.

When a new thread is created then it will have its own stack separate from the main() call stack.

[View All Answers](#)

Question - 7:

What is starvation?

Ans:

Starvation is a situation when some threads acquired the shared resources for long time and therefore other threads are not able to access those resources and not able to do anything further.

For example, suppose an object provides a synchronized method that often takes a long time to return.

If one thread invokes this method frequently, other threads that also require frequent synchronized access to that object will be blocked.

In Java, Starvation can be caused by inappropriate allocation of thread priorities.

A thread with low priority can be starved by the threads of higher priority if the higher priority threads do not release shared resources time to time.

[View All Answers](#)

Question - 8:

Explain deadlock?

Ans:

When two or more threads are waiting for each other and get blocked forever then this situation is called Deadlock.

During deadlock two threads, each thread has acquired a lock on one resource and trying to acquire a lock on resource of other thread.

Each thread will wait indefinitely for the other to release the lock, unless one of the user processes is stopped.

In terms of Java API, thread deadlock situation can arise in following conditions:

1. When two threads call Thread.join() on each other.
2. When two threads use nested synchronized blocks to lock two objects and the blocks lock the same objects in different order.

[View All Answers](#)

Question - 9:

How to debug our application for issues when multiple threads are being executed?

Ans:

Following are some way to debug issues in multi-threaded applications in Java.

-By using logging and print statements along with thread names. In this way we can know about the flow of thread execution.

-With the use of debugging functionality available in Eclipse and JDeveloper.

-We can write a thread dump of the application which will give the information about the active threads at a point of time.

This is most effective way for detecting deadlocks in production systems.

[View All Answers](#)

Question - 10:

What is thread Local variable?

Ans:

If a variable is declared as threadLocal then all the thread which accesses that variable will maintain its own copy of variable and would not use the copy of any other thread.

For example let consider the scenario of giving JDBC connection to each thread.

```
public class ThreadLocal {
    public Object get();
    public void set(Object myValue);
    public Object FirstValue();
}
```

Implementation of ThreadLocal

```
public class ConnectionDispenser {
    private static class ThreadLocalConnection extends ThreadLocal {
        public Object FirstValue () {
            return DriverManager.getConnection(ConfigurationSingleton.getDbUrl());
        }
    }
    private static ThreadLocalConnection con = new ThreadLocalConnection();
    public static Connection MyConnection() {
        return (Connection) con.get();
    }
}
```



```
}  
}
```

[View All Answers](#)

Question - 11:

What are the methods of the thread class used to schedule the threads?

Ans:

The methods of the thread class used to schedule the threads are as follows:

```
-public final void join() throws InterruptedException  
-public final void notify()  
-public final void notifyAll()  
-public static void yield()  
-public final void setPriority(int priority)  
-public static void sleep(long millis) throws InterruptedException  
-public final void wait() throws InterruptedException
```

[View All Answers](#)

Question - 12:

Explain the method of Runnable interface with example?

Ans:

In this method of creating thread, we have to implement the Runnable interface and implement the run() method in our class.

We have to create an object of our class.

Then we you have to pass the reference of that object for creating a new object of Thread

Invoke the start method using this Thread object which will create a new thread of execution.

For example

```
public class MyThread implements Runnable{  
    public void run(){  
        // code to execute under the thread  
    }  
    public static void main(String [] args){  
        MyThread c = new NewThread();  
        Thread t = new Thread(c);  
        t.start();  
    }  
}
```

[View All Answers](#)

Question - 13:

What are the advantages or usage of threads in Java?

Ans:

Following are the advantage of using threads in Java.

Threads support concurrent operations. For example, Server can handle multiple requests coming from different clients by managing separate thread for each request.

Threads often result in simpler programs. Updating of separate views can be managed by separate Thread to give continuous updates.

Threads provide a high degree of control on application.

Threading gives the concurrency in our application by simplified coding.

Using threading, a computer with more than one CPU can execute multiple threads on different functional units without using time sharing.

The cost of communication between threads is relatively low.

[View All Answers](#)

Question - 14:

What happens if we invoke run method without calling the start method for a thread instance?

Ans:

If we instantiate a thread it is called in new state until the Start() method is called.

If we don't call a start() method for that thread instance, the thread is not called alive.

If we invoke run method without calling the start method for a thread instance, the code in run() method wil not be executed by a new thread but it will be executed by the existing thread only.

[View All Answers](#)

Question - 15:

What is difference between thread and process?

Ans:

Process has its own memory address while the thread share the address of the process by which it is created.

Thread can access the data segment of its process directly while processes can access their own copy of the data segment of its parent process.

Threads have almost no overheads while processes may have overheads.

We can create a new thread easily but to create new process we have to duplicate the parent process.

If any changes in main thread are occurred it can affect the behavior of the other threads of the same process. While in case of process if any changes is occurred in parent process it won't affect the behavior of parent process.

The direct communication is possible between threads of same process while in case of same sibling processes only interprocess communication is possible.

[View All Answers](#)



Question - 16:

What are the different states of a threads lifecycle?

Ans:

Following are the different states of threads:

1. New - A thread is called in New state until the start() method is called for the object of that thread. The thread is not alive in this state.
2. Runnable - The thread is called in Runnable state after the start() method is called for the object of that thread. The thread may enter into the Runnable state from Running state. The thread is called alive in this state.
3. Running - The thread is called in Running state when the thread scheduler take it from the Runnable thread's pool.
4. Waiting/Blocked/Sleeping - In these states the thread is alive but it's not in Runnable state. A running thread is called in these states after wait() or sleep() method is called that thread or when the thread is blocked while waiting for I/O resources.
5. Dead - After the execution of run() method is complete then the thread is called in dead state. Once a thread is dead then it cannot be start again.

[View All Answers](#)

Question - 17:

Explain the method of Thread class with example?

Ans:

In this method of creating thread, we have to extend the Thread class and override the run() method in our class to create a Thread.

We have to create an object of the our class.

Once the object is created then we have to invoke the start() method and it will generate a new thread of execution.

For example

```
public class MyThread extends Thread{
public void run(){
// code to execute under the thread
}
public static void main(String [] args){
MyThread c = new MyThread();
c.start();
}
}
```

[View All Answers](#)

Question - 18:

What is Thread leak?

Ans:

Thread leak is when application does not release references of the thread object and those threads do not get garbage collected.

Number of such unused threads increases with time and it can cause issues in the application like long response time.

To overcome this problem we can do the following

1. By maintaining a log for all entry and exit point of thread.
2. Check how the new thread is created and how it is closed.
3. By using exception handling etc.

[View All Answers](#)

Question - 19:

What is daemon thread and how it differs from user thread?

Ans:

A user thread is created by the application (user) while the daemon thread is created by the JVM to provide services to user threads.

If user threads are exists then only daemon threads exists.

You can make a user thread as daemon thread by using setDaemon(true) method.

In an application when only daemon threads are running (i.e. no user thread is running) then JVM will shut down the application and subsequently stops all daemon threads.

To keep the application running, at least one user thread should be running.

[View All Answers](#)

Question - 20:

Explain volatile?

Ans:

When more than one threads access the same variable then each thread will maintain its own copy of that variable in the form of local cache.

When a thread is changing the value of variable, it is actually changing the local cache not the main variable memory.

Every thread which is using the same variable doesn't know anything about the values changed by another thread.

To overcome this problem, if you declare a variable as volatile, then whenever a thread is changing the value of a volatile, it is actually changing to the main memory. So, all other threads which are accessing the same variable can get the changed value.

[View All Answers](#)

Question - 21:

Name the four states associated in the thread?

Ans:

new,
runnable,
blocked,
dead



[View All Answers](#)

Question - 22:

How many threads at a time can access a monitor?

Ans:

one

[View All Answers](#)

Question - 23:

Name the default thread at the time of starting the program?

Ans:

main thread

[View All Answers](#)

Question - 24:

Name the method for setting the priority?

Ans:

setPriority()

[View All Answers](#)

Question - 25:

Provide the values for max-priority, min-priority and normal-priority level?

Ans:

10,

1,

5

[View All Answers](#)

Question - 26:

What is the sleep() method Data type for the parameter?

Ans:

long

[View All Answers](#)

Question - 27:

Name the mechanism defined by java for the Resources to be used by only one Thread at a time?

Ans:

Synchronisation

[View All Answers](#)

Question - 28:

Name the methods used for Inter Thread communication?

Ans:

wait(),

notify() &

notifyall()

[View All Answers](#)

Question - 29:

Write the signature of the constructor of a thread class?

Ans:

```
Thread(Runnable threadobject,String threadName)
```

```
{
```

```
}
```

[View All Answers](#)

Question - 30:

Name the methods available in the Thread class?

Ans:

isAlive(),

join(),

resume(),



suspend(),
stop(),
start(),
sleep(),
destroy()

[View All Answers](#)

Question - 31:

Name the methods available in the Runnable Interface?

Ans:

run()

[View All Answers](#)

Question - 32:

What is time slicing?

Ans:

Timeslicing is the method of allocating CPU time to individual threads in a priority schedule.

[View All Answers](#)

Question - 33:

Difference between Process and Thread?

Ans:

A process can contain multiple threads.

A process has its own memory address space whereas a thread doesn't.

Threads share the heap belonging to their parent process.

One process cannot corrupt another process

A thread can write the memory used by another thread.

[View All Answers](#)

Question - 34:

What are the two types of multitasking?

Ans:

Two types of multitasking are Process-based and Thread-based

[View All Answers](#)

Question - 35:

Explain Preemptive scheduling and time slicing?

Ans:

In preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence.

In time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks.

The scheduler then determines which task should execute next, based on priority and other factors.

Briefly explain daemon thread.

Daemon thread is a low priority thread which runs in the background performs garbage collection operation for the java runtime system.

[View All Answers](#)

Java Programing Most Popular & Related Interview Guides

- 1 : [Core Java Interview Questions and Answers.](#)
- 2 : [Hibernate Interview Questions and Answers.](#)
- 3 : [IBM WebSphere Interview Questions and Answers.](#)
- 4 : [Advanced Java Interview Questions and Answers.](#)
- 5 : [Spring Framework Interview Questions and Answers.](#)
- 6 : [Full Stack Developer \(Java\) Interview Questions and Answers.](#)
- 7 : [JSF Interview Questions and Answers.](#)
- 8 : [JDBC Interview Questions and Answers.](#)
- 9 : [Java Swing Programming Interview Questions and Answers.](#)
- 10 : [Java JSP Programming Interview Questions and Answers.](#)

Follow us on FaceBook

www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)