What are final methods of Object class ?

Ans -> Wait notify and notifyAll method

What are the disadvantages of Hibernate ?

What is difference between Synchronized Hashmap and ConcurrentHashMap?

If we have configured Second level cache and data is changed from Database ? what output will be returned on calling get or load method?

What is Inverse keyword/Annotation In Hibernate?

How different annotation of Spring interact to each other?

Can We write any other annotation instead of @Contoller?

What are different view resolver?

What are the different Bean Scope of Spring

What is dependency injection? What happens internally in case of using @Autowire annotation

What is wildcard? Can we use any other keyword instead of extends in case of WildCard?

What is the use of Ribbon and Zuul Proxy ? How Hystrix comes in picture?

How you secure your Rest Webservice in your application?

Can We throw Error object using Throws Clause in catch block?

What is difference between war and jar?

What is default logging mechanism of Spring boot?

What is difference between Callable and Runnable?

What is difference between KeySet nad Hashset?

What are Rest API? What are different methods of Restful Api?

What is difference between Put and POST?

When Singleton Design Pattern fails?

What is double locking mechanism?

What is the use of intern() method?

What is Connection pooling? Which API do you use for connection pooling in your application and How?

What is difference between @ Bean and @Component and @Service?

What is Transaction management in JPA or Spring?

Different annotations in Spring and its explanation?

What is the way to call default and static method call in Java 8?

What is difference between @ Component and @ Service?

What is difference between Synchronized Hashmap and ConcurrentHashMap?

**What are the different ways to iterate a HashMap?**

If you're only interested in the keys, you can iterate through the [keySet()](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Map.html#keySet%28%29) of the map:

Map<String, Object> map = ...;

for (String key : map.keySet()) {

// ...

}

If you only need the values, use [values()](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Map.html#values%28%29):

for (Object value : map.values()) {

// ...

}

Finally, if you want both the key and value, use [entrySet()](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Map.html#entrySet%28%29):

for (Map.Entry<String, Object> entry : map.entrySet()) {

String key = entry.getKey();

Object value = entry.getValue();

// ...

}

What Is Entry Interface implementation?

[Home](https://www.concretepage.com/)  >  [Core Java Interview Q & A](https://www.concretepage.com/interview/java-interview)

Core Java Interview Questions: Executor Framework

January 03, 2014

Qns-1: What do you understand by Executor Framework in Java?

Ans: Executor Framework in java has been introduced in JDK 5. Executor Framework handles creation of thread, creating the thread pool and checking health while running and also terminates if needed.

Qns-2: What is the role of ExecutorService in Java?

Ans: ExecutorService provides different methods to start and terminate thread. There are two methods execute() and submit() in ExecutorService. Execute() method is used for threads which is Runnable and submit() method is used for Callable threads.

Qns-3: What is Executors in java Executor Framework?

Ans: Executors is a factory that provides the methods to return ExecutorService, ScheduledExecutorService, ThreadFactory. Find some method details.   
**newFixedThreadPool()**: It returns the pool with fixed number of size. We need to pass the number of threads to this method. If concurrently task are submitted more than the pool size, then rest of task need to wait in queue. It returns ExecutorService.   
**newScheduledThreadPool**: This also creates a fixed size pool but it can schedule the thread to run after some defined delay. It is useful to schedule the task. It returns ScheduledExecutorService.   
**newCachedThreadPool()**: There is no fixed size of this pool. Thread will be created at run time and if there is no task it will alive for 60 second and then die. For short lived threads this pool works good. It returns ExecutorService.

Qns-4: What is the role of FutureTask and Future in java?

Ans: FutureTask is a cancellable asynchronous computation in java. It can cancel the task which is running. Once the FutureTask will be cancelled, it cannot be restarted. Future is result of asynchronous computation. Future checks if task is complete and if completed it gets the output.

Qns-5: What is difference between shutdownNow() and shutdown() in Executor Framework in java?

Ans: shutdown() and shutdownNow() methods belongs to ExecutorService. shutdown() method tries to stop the threads and do not accept new task to execute but it completes the execution which has been submitted. shutdownNow() methods also tries to stop the running threads and will not execute any task which has been submitted but not started.

Qns-6: How to terminate a thread in Executor Framework in java?

Ans: ExecutorService provides a method awaitTermination(long timeout, TimeUnit unit) that takes time and unit of time as an arguments. After that time thread pool is terminated. Suppose we need to terminate a task just now, then we can do as

ExecutorService.awaitTermination(0, TimeUnit.SECONDS)

Qns-7: What is the role of Executors.privilegedThreadFactory() in Executor Framework?

Ans: privilegedThreadFactory returns a thread factory that creates thread with same permission as main thread.

Qns-8: What is the role of Executors.unconfigurableExecutorService in Executor Framework?

Ans: unconfigurableExecutorService returns an object that delegates all methods of ExecutorService to the given executor so that any other method cannot accessed by cast.

Qns-9: What are the different policy in Executor Framework?

Ans: There are different policy within ThreadPoolExecutor in java.   
**a. ThreadPoolExecutor.AbortPolicy**: AbortPolicy is a handler for rejected task. It handles those task which has been rejected.   
**b. ThreadPoolExecutor.CallerRunsPolicy**: This also handles the rejected task and runs the rejected task directly.   
**c. ThreadPoolExecutor.DiscardOldestPolicy**: This handles those rejected task that is oldest and unhandled. It discards those that oldest task.   
**d. ThreadPoolExecutor.DiscardPolicy**: This is the handler for those rejected task that are rejected silently.

Qns-10: How to get return value of a callable thread in java Executor Framework?

Ans: Using Future, we can get the return value of callable thread.

ExecutorService exService = Executors.newCachedThreadPool();

Future<Integer> future=exService.submit(new CallableThread());

int val=future.get();