Java Interview

# Java Core

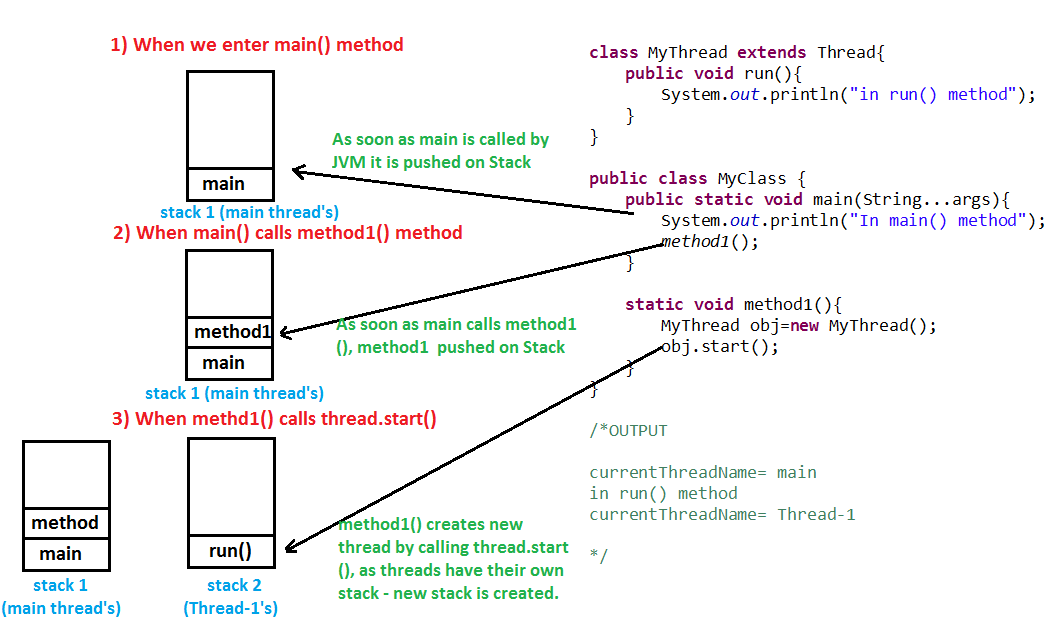
## Multithreading, concurrency and thread basics

1. What is Thread in Java?

* Threads consumes CPU in best possible manner, hence enables multi processing. Multi threading reduces idle time of CPU which improves performance of application.
* Thread are light weight process.
* A thread class belongs to java.lang package.
* We can create multiple threads in java, even if we don’t create any Thread, one Thread at least do exist i.e. main thread.
* Multiple threads run parallelly in java.
* Threads have their own stack.
* Advantage of Thread : Suppose one thread needs 10 minutes to get certain task, 10 threads used at a time could complete that task in 1 minute, because threads can run parallely.

1. What is difference between Process and Thread in java?
2. Does Thread implements their own Stack, if yes how? (Important)

Yes, [Threads have their own stack](http://www.javamadesoeasy.com/2015/03/threads-implement-their-own-stack.html). This is very interesting question, where interviewer tends to check your basic knowledge about how [threads internally maintains their own stacks](http://www.javamadesoeasy.com/2015/03/threads-implement-their-own-stack.html). I’ll be explaining you the concept by diagram.



1. What is the difference between a Thread and an Executor?

Creating a running thread in Java is an expensive operation, and an operating system may limit the number of threads provided to a running application at any one time. By using a thread pool, you can create threads when needed, and, rather than creating new threads, you can reuse a thread after it has completed running previous code.

1. How can you ensure all threads that started from main must end in order in which they started and also main should end in last? (Important)

We can use [join() method](http://www.javamadesoeasy.com/2015/03/join-method-ensure-all-threads-that.html) to ensure all threads that started from main must end in order in which they started and also main should end in last.In other words waits for this thread to die. Calling join() method internally calls join(0);

<https://www.journaldev.com/1024/java-thread-join-example>

1. What is significance of using [Volatile](http://www.javamadesoeasy.com/2015/03/volatile-keyword-in-java-difference.html) keyword? (Important)

If a field is declared [volatile](http://www.javamadesoeasy.com/2015/03/volatile-keyword-in-java-difference.html), in that case the Java memory model ensures that all threads see a consistent value for the variable.

<http://www.javamadesoeasy.com/2015/03/volatile-keyword-in-java-difference.html>

1. Why wait(), notify()  and notifyAll() are in Object class and not in Thread class? (Important)

Every Object has a monitor, acquiring that monitors allow thread to hold lock on object.

wait(), notify() and notifyAll() are called on objects only > When wait() method is called on object by thread it waits for another thread on that object to release object monitor by calling [notify() or notifyAll()](http://www.javamadesoeasy.com/2015/03/difference-between-notify-and-notifyall.html) method on that object.

1. What is ThreadLocal?

Java ThreadLocal is used to create thread-local variables. We know that all threads of an Object share it’s variables, so if the variable is not thread safe, we can use synchronization but if we want to avoid synchronization, we can use ThreadLocal variables.  
Every thread has it’s own ThreadLocal variable and they can use it’s get() and set() methods to get the default value or change it’s value local to Thread. ThreadLocal instances are typically private static fields in classes that wish to associate state with a thread. Check this post for small example program showing [ThreadLocal Example](https://www.journaldev.com/1076/java-threadlocal-example).

<https://www.journaldev.com/1076/java-threadlocal-example>

1. Question 20. What is [deadlock](http://www.javamadesoeasy.com/2015/03/deadlock-in-multithreading-program-to.html) in multithreading? Write a program to form [DeadLock](http://www.javamadesoeasy.com/2015/03/deadlock-in-multithreading-program-to.html)in multi threading and also how to solve DeadLock situation. What measures you should take to avoid deadlock? (Important)
2. What are differences between wait and sleep method in Java?

The only major difference is to wait for release the lock or monitor while sleep doesn't release any lock or monitor while waiting. The wait is used for inter-thread communication while sleep is used to introduce pause on execution.

<http://javarevisited.blogspot.sg/2011/12/difference-between-wait-sleep-yield.html>

1. Difference between ExecutorService.submit() and Executor.execute() methods in Java?

<https://javarevisited.blogspot.it/2016/04/difference-between-ExecutorServie-submit-vs-Executor-execute-method-in-Java.html>

1. dsad

Ref: <https://javarevisited.blogspot.it/2011/07/java-multi-threading-interview.html>

## Java collections framework