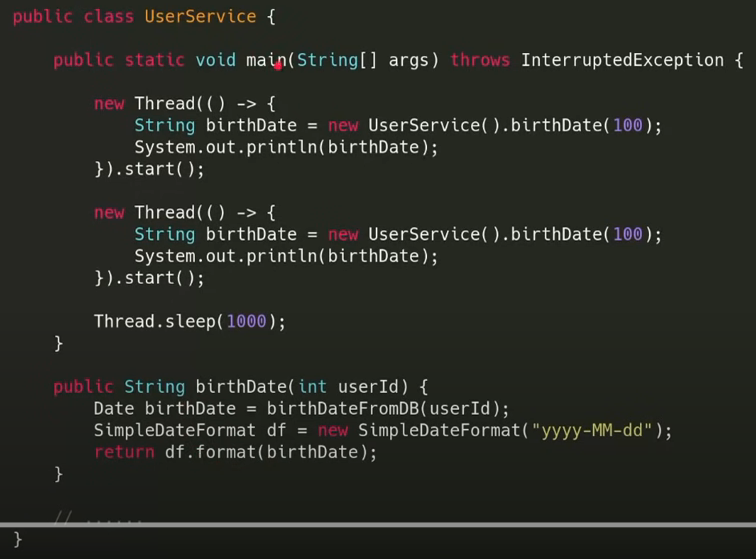
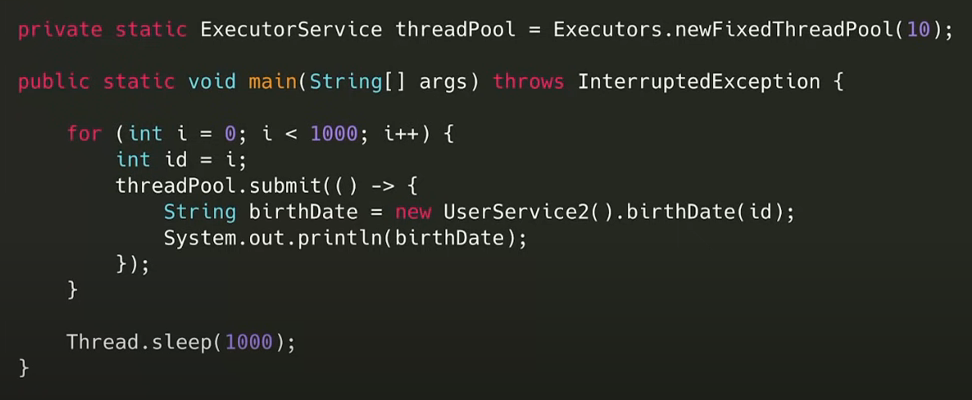
**What is ThreadLocal in Java?**

ThreadLocal is a special class in Java that helps us achieve thread safety by providing per-thread contexts and maintaining them separately for each thread. In other words, ThreadLocal is a Java class that can be used to define variables accessible solely by the thread that created them. This can be useful in a number of situations, but the most common use case is when you need to store data that should not be shared between threads.

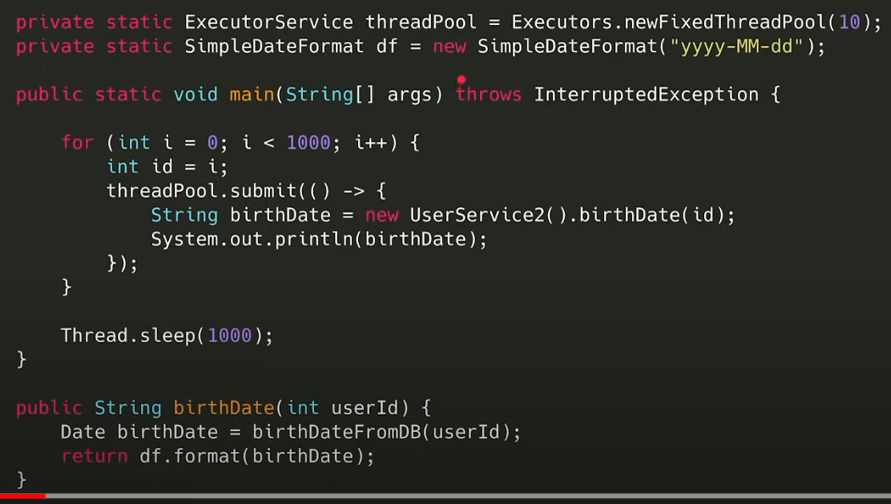
* According to below code, both thread will internally create the SimpleDateFormat object in heap memory so after stating both the thread there will be two object present in heap memory.



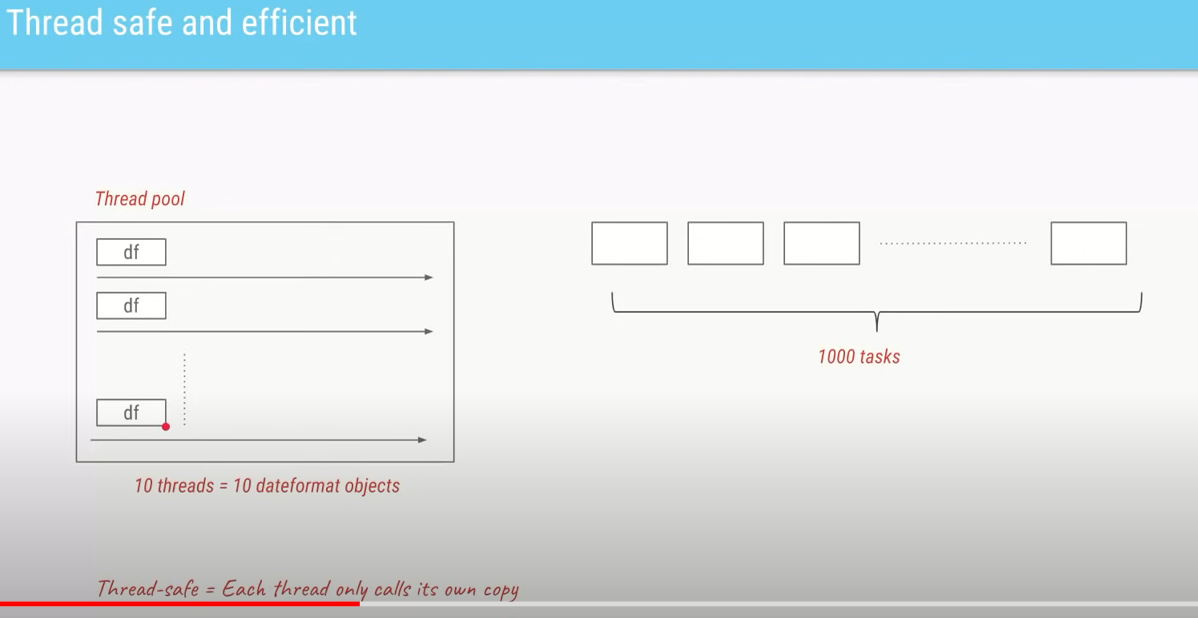
* So far it looks good, but if we change the code little bit and added a for loop to create 1000 thread like below.



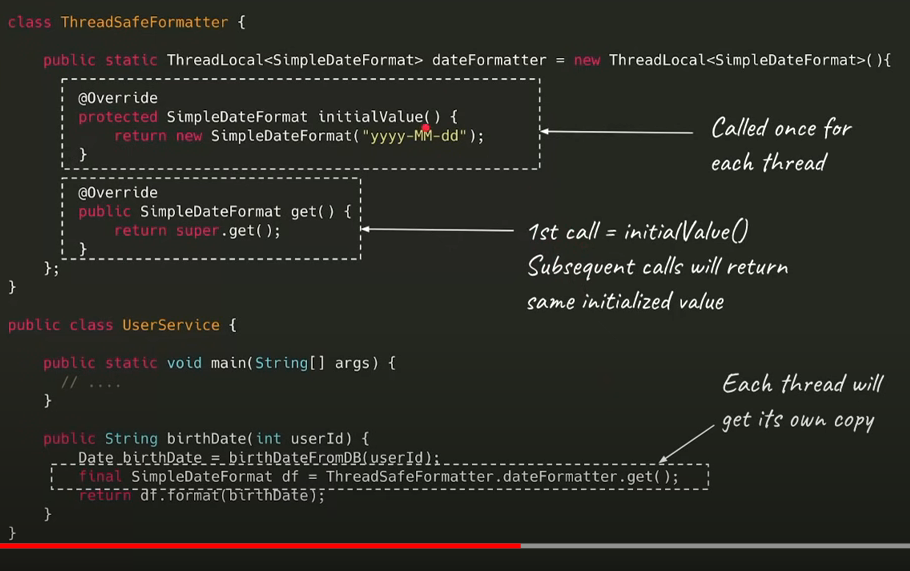
* In this it is impractical to create 1000s of the SimpleDateFormat object in memory which will consume lot of memory and can cause the memory issue in JVM.
* So one way is to resolve this memory issue is to create a global(class level) SimpleDateFromt object so that it can be reused each time.



* But the problem with SimpleDateFormat class is, it is thread safe so there will be data integrity and data insistency issue.
* One way to resolve the issue is having locks. However, if we have locks that means only one threat is allowed to access SimpleDateFormat object.
* Here we have introduce synchronization but due this we will face the perfomce issue as only one thread will be allowed to access the object.
* That means we cannot have 1 object for per thread which wee cause the memory issue and we also cannot create global object across all the thread as it will be not thread safe there will be data inconsistency issue.
* And if we try to have locks(syncronization) on object, there will be performance issue.
* So what we want is to have objects in in threadpool for each thread and the same object will be used for the next bunch of threads.
* ThreadLoacl does the same thing for use like below.







* ThreadLocal has two methods
* initialValue() which will get call at the start of the thread only.
* Get() if it is a first time call the get() internally calls the initialValue() and subsequent call will execute the get() body only.
* If we use the java8 then all the boilerplace caode will get deleted as below

