```java
-------------ThreadLocalTask.java---------
package com.ameya.tasks;

import java.util.Date;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.atomic.AtomicInteger;

public class ThreadLocalTask implements Runnable {
    private static final AtomicInteger nextId=new AtomicInteger(0);
    private static final ThreadLocal<Integer> threadId=new
    ThreadLocal<Integer>() {
        @Override
        protected Integer initialValue() {
            return nextId.getAndIncrement();
        }
    };
    public int getThreadId() {
        return threadId.get();
    }
    private static final ThreadLocal<Date> startDate=new
    ThreadLocal<Date>() {
        protected Date initialValue() {
            return new Date();
        }
    };
    @Override
    public void run() {
        System.out.printf("Starting Thread : %s :
        %s\n",getThreadId(),startDate.get());
        try {
            TimeUnit.SECONDS.sleep((int)Math.rint(Math.random()*10));
        }catch(InterruptedException e) {
            e.printStackTrace();
        }
        System.out.printf("Thread Finished : %s :
        %s\n",getThreadId(),startDate.get());
    }

}
--------------TestThreadLocal.java---------
package com.ameya.test;
```

```java

import com.ameya.tasks.ThreadLocalTask;

public class TestThreadLocal {

    public static void main(String[] args) {
        new Thread(new ThreadLocalTask()).start();
        try {
            Thread.sleep(2000);
        }catch(InterruptedException e) {
            e.printStackTrace();
        }
        new Thread(new ThreadLocalTask()).start();
        try {
            Thread.sleep(2000);
        }catch(InterruptedException e) {
            e.printStackTrace();
        }
        new Thread(new ThreadLocalTask()).start();

    }

}
```