

```

1  -----TaskForCachedThreadPool.java-----
2  package com.ameya.tasks;
3
4  import java.util.concurrent.TimeUnit;
5
6  public class TaskForCachedThreadPool implements Runnable {
7
8      @Override
9      public void run() {
10         Long num=(long)(Math.random()/30);
11         System.out.println("Thread Name ::
12         "+Thread.currentThread().getName());
13         try {
14             TimeUnit.SECONDS.sleep(num);
15         } catch (InterruptedException e) {
16             e.printStackTrace();
17         }
18         System.out.println("After Sleep Thread Name ::
19         "+Thread.currentThread().getName());
20     }
21 }
22 -----TestCachedThreadPoolExecutor.java-----
23 -
24 package com.ameya.test;
25
26 import java.util.concurrent.ExecutorService;
27 import java.util.concurrent.Executors;
28 import java.util.concurrent.ThreadFactory;
29 import java.util.concurrent.ThreadPoolExecutor;
30
31 import com.ameya.tasks.TaskForCachedThreadPool;
32
33 public class TestCachedThreadPoolExecutor {
34
35     public static void main(String[] args) {
36         ExecutorService executor=Executors.newCachedThreadPool();
37         //ThreadFactory
38         threadFactory=Executors.defaultThreadFactory();
39         //ExecutorService

```

```
38     executor=Executors.newCachedThreadPool(threadFactory);
39     ThreadPoolExecutor myPool=(ThreadPoolExecutor)executor;
40     System.out.println("Size Of My Pool :: "+myPool.getPoolSize());
41     executor.submit(new TaskForCachedThreadPool());
42     executor.submit(new TaskForCachedThreadPool());
43     System.out.println("Total Threads Scheduled ::
44     "+myPool.getTaskCount());
45     executor.shutdown();
46 }
47
48
```