

### Task 5: Thread Pools and Concurrency Utilities

Create a fixed-size thread pool and submit multiple tasks that perform complex calculations or I/O operations and observe the execution.

ANS:

```
package com.Day23;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
public class Task5 {
    public static void main(String[] args) {
        // Create a fixed-size thread pool with 5 threads
        ExecutorService executor = Executors.newFixedThreadPool(5);
        // Submit multiple tasks to the thread pool
        for (int i = 1; i <= 10; i++) {
            Runnable task = new Task("Task " + i);
            executor.submit(task);
        }
        // Shutdown the thread pool after all tasks are completed
        executor.shutdown();
    }
    static class Task implements Runnable {
        private final String name;
        public Task(String name) {
            this.name = name;
        }
        @Override
        public void run() {
            System.out.println(name + " is starting...");
            // Simulate some complex calculation or I/O operation
            try {
                Thread.sleep(2000); // Simulating a task that takes 2 seconds to complete
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println(name + " is done.");
        }
    }
}
```