Task 7: Writing Thread-Safe Code, Immutable Objects

Design a thread-safe Counter class with increment and decrement methods. Then demonstrate its usage from multiple threads. Also, implement and use an immutable class to share data between threads

```
package Day_18;


class Counter {
    private int count = 0;

    public synchronized void increment() {
        count++;
    }

    public synchronized void decrement() {
        count--;
    }

    public synchronized int getCount() {
        return count;
    }
}

final class ImmutableData {
    private final String data;

    public ImmutableData(String data) {
        this.data = data;
    }

    public String getData() {
        return data;
    }
}

public class Task7 {
    public static void main(String[] args) {
        // Create a Counter instance
        Counter counter = new Counter();

        // Create and start multiple threads to increment and decrement the counter
        Thread[] threads = new Thread[10];
        for (int i = 0; i < threads.length; i++) {
            threads[i] = new Thread(() -> {
                for (int j = 0; j < 1000; j++) {
                    counter.increment();
                    counter.decrement();
                }
            });
            threads[i].start();
        }

        // Wait for all threads to finish
        for (Thread thread : threads) {
            try {
                thread.join();
            } catch (InterruptedException e) {
                e.printStackTrace();
```

```
            }
        }

        // Print the final count
        System.out.println("Final count: " + counter.getCount());

        // Immutable data shared between threads
        ImmutableData immutableData = new ImmutableData("Shared Immutable Data");

        // Create and start threads to access immutable data
        Thread[] immutableThreads = new Thread[5];
        for (int i = 0; i < immutableThreads.length; i++) {
            immutableThreads[i] = new Thread(() -> {
                System.out.println("Immutable Data: " + immutableData.getData());
            });
            immutableThreads[i].start();
        }
    }
}
```