DAY 23:

ASSIGNMENT 1:

Task 1: Creating and Managing Threads

Write a program that starts two threads, where each thread prints numbers from 1 to 10 with a 1-second delay between each number.

```
class NumberPrinterThread extends Thread {
    private String threadName;

    public NumberPrinterThread(String threadName) {
        this.threadName = threadName;
    }

    @Override
    public void run() {
        try {
            for (int i = 1; i <= 10; i++) {
                System.out.println(threadName + ": " + i);
                Thread.sleep(1000); // Delay for 1 second
            }
        } catch (InterruptedException e) {
            System.out.println(threadName + " interrupted.");
        }
    }

    public static void main(String[] args) {
```

```
        // Create two NumberPrinterThread objects
        NumberPrinterThread t1 = new NumberPrinterThread("Thread 1");
        NumberPrinterThread t2 = new NumberPrinterThread("Thread 2");

        // Start the threads
        t1.start();
        t2.start();

        // Wait for both threads to finish
        try {
            t1.join();
            t2.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        System.out.println("Both threads have finished.");
    }
}
```

Explanation:

1. *NumberPrinterThread Class*: Extends the Thread class.
   - threadName is used to distinguish between the two threads.
   - The run method contains a loop that prints numbers from 1 to 10 with a 1-second delay using Thread.sleep(1000).

2. *Main Method*:
   - Creates two instances of NumberPrinterThread, each with a unique thread name.
   - Starts both threads using the start() method.

- Uses join() method to wait for both threads to complete before printing a final message indicating that both threads have finished.

Running the Program:

When you run this program, the output will show interleaved numbers from both threads, each printing from 1 to 10 with a delay of 1 second between each number. This approach demonstrates how to create threads by extending the Thread class, providing an alternative to implementing the Runnable interface.