ANSH PANDEY

2300290130036

IT – (A) - 36

LAB – 5

5.1 WAP to Create a class MyThread derived from Thread class and override the run method. Create a class ThreadDemo having a main method. Create 2 objects of MyThread class and observe the behavior of threads.

```
// MyThread.java
class MyThread extends Thread {
    private String threadName;

    MyThread(String name) {
        threadName = name;
    }

    @Override
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(threadName + " - Count: " + i);
            try {
                Thread.sleep(500); // Pause for half a second
            } catch (InterruptedException e) {
                System.out.println(threadName + " interrupted.");
            }
        }
        System.out.println(threadName + " finished.");
    }
}

public class ThreadDemo {
    public static void main(String[] args) {

        MyThread thread1 = new MyThread("Thread-1");
        MyThread thread2 = new MyThread("Thread-2");
        thread1.start();
        thread2.start();
```

```
        try {
            thread1.join();
            thread2.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        System.out.println("Main thread exiting.");
    }
}
```

```
PS D:\javalab> cd week_5
PS D:\javalab\week_5> javac ThreadDemo.java
PS D:\javalab\week_5> java ThreadDemo
Thread-1 - Count: 1
Thread-2 - Count: 1
Thread-1 - Count: 2
Thread-2 - Count: 2
Thread-1 - Count: 3
Thread-2 - Count: 3
Thread-1 - Count: 4
Thread-2 - Count: 4
Thread-1 - Count: 5
Thread-2 - Count: 5
Thread-1 finished.
Thread-2 finished.
Main thread exiting.
PS D:\javalab\week_5>
```

5.2 WAP to Modify the above to create MyThread class by implementing Runnable interface and observe the behavior of threads.

```
// MyThread.java
class MyThread extends Thread {
    private String threadName;

    MyThread(String name) {
        threadName = name;
    }

    @Override
    public void run() {
```

```
        // Run method logic
        for (int i = 1; i <= 5; i++) {
            System.out.println(threadName + " - Count: " + i);
            try {
                Thread.sleep(500); // Pause for half a second
            } catch (InterruptedException e) {
                System.out.println(threadName + " interrupted.");
            }
        }
        System.out.println(threadName + " finished.");
    }
}

public class MyThreadProgram {
    public static void main(String[] args) {
        MyThread runnable1 = new MyThread("Thread-1");
        MyThread runnable2 = new MyThread("Thread-2");

        Thread thread1 = new Thread(runnable1);
        Thread thread2 = new Thread(runnable2);
        thread1.start();
        thread2.start();
        try {
            thread1.join();
            thread2.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        System.out.println("Main thread exiting.");
    }
}
```

```
PS D:\javalab\week_5> javac MyThreadProgram.java
PS D:\javalab\week_5> java MyThreadProgram
Thread-2 - Count: 1
Thread-1 - Count: 1
Thread-1 - Count: 2
Thread-2 - Count: 2
Thread-2 - Count: 3
Thread-1 - Count: 3
Thread-2 - Count: 4
Thread-1 - Count: 4
Thread-2 - Count: 5
Thread-1 - Count: 5
Thread-2 finished.
Thread-1 finished.
Main thread exiting.
PS D:\javalab\week_5>
```