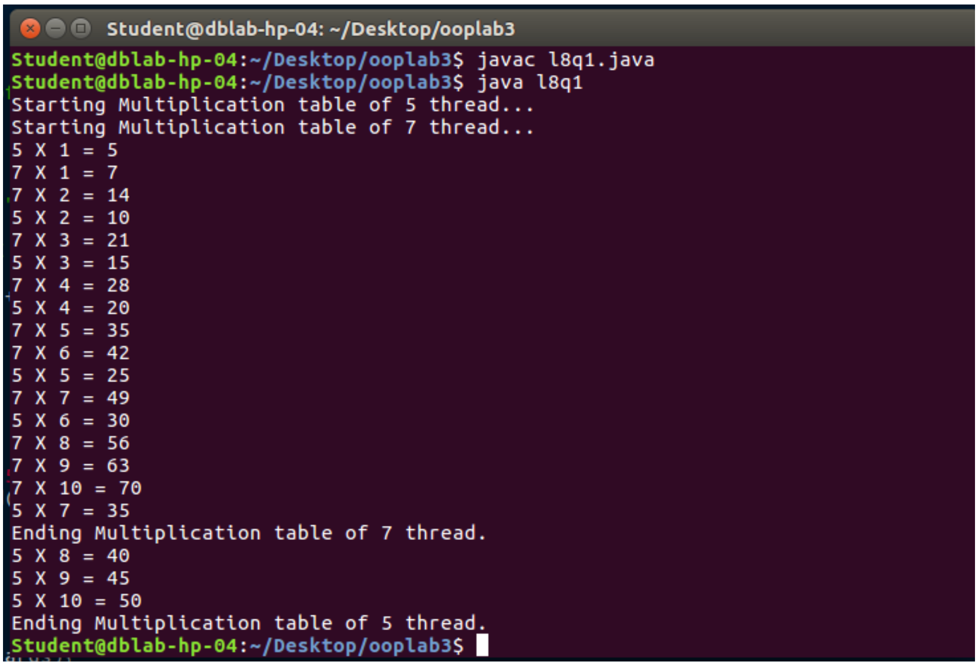
Ayush Goyal
190905522

OOP Lab 8 – Multithreading (Session 3)

Q1)

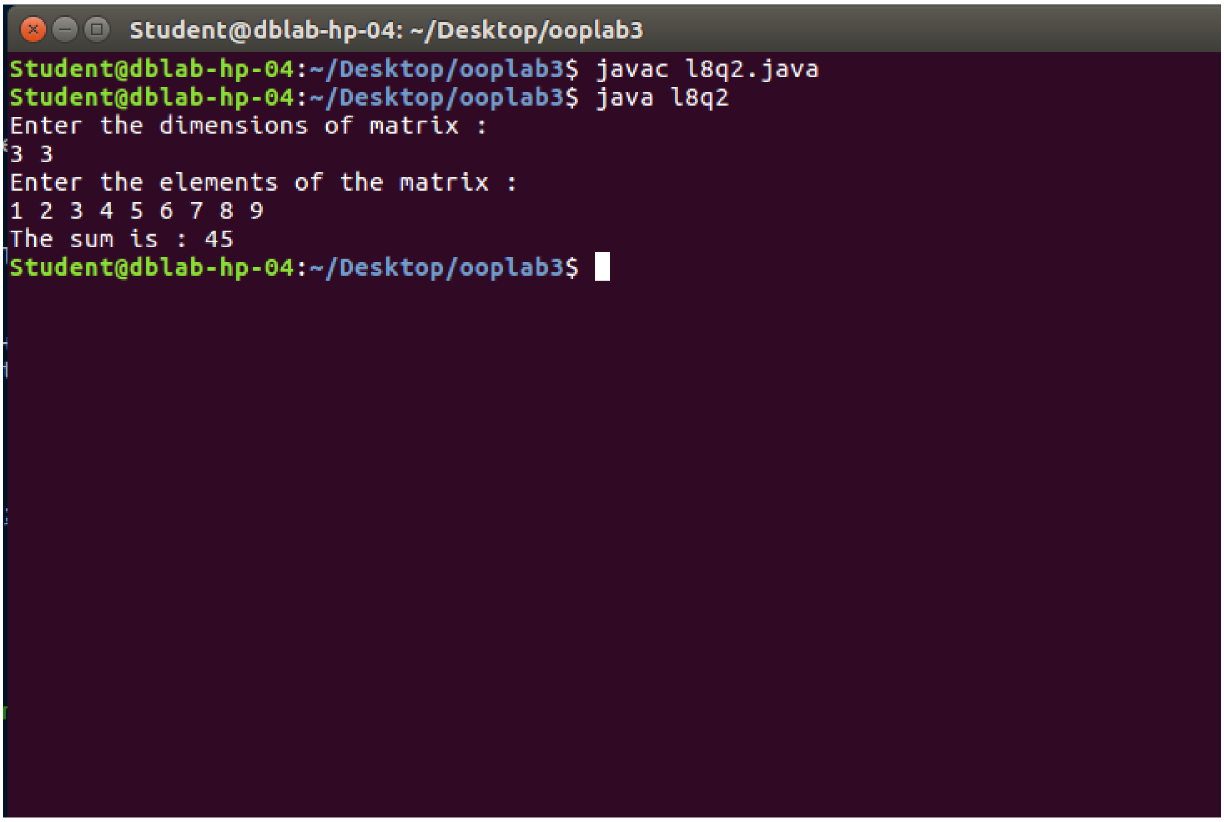
```
import java.util.*;

class MyThread extends Thread{
    int n;
    MyThread(int n){
        super("Multiplication table of " + n);
        this.n = n;
        start();
    }
    public void run(){
        System.out.println("Starting " + this.getName() + " thread...");
        for(int i = 1; i<= 10; i++){
            System.out.println(n +" X " + i + " = " + n*i);
        }
        System.out.println("Ending " + this.getName() + " thread.");
    }
}

class Tables{
    Tables(){
        MyThread five = new MyThread(5);
        MyThread seven = new MyThread(7);
    }
}

 public class l8q1{
    public static void main(String[] args){
        Tables newtables = new Tables();
    }
}
```



```
Student@dblab-hp-04: ~/Desktop/ooplab3
Student@dblab-hp-04:~/Desktop/ooplab3$ javac l8q1.java
Student@dblab-hp-04:~/Desktop/ooplab3$ java l8q1
Starting Multiplication table of 5 thread...
Starting Multiplication table of 7 thread...
5 X 1 = 5
7 X 1 = 7
7 X 2 = 14
5 X 2 = 10
7 X 3 = 21
5 X 3 = 15
7 X 4 = 28
5 X 4 = 20
7 X 5 = 35
7 X 6 = 42
5 X 5 = 25
7 X 7 = 49
5 X 6 = 30
7 X 8 = 56
7 X 9 = 63
7 X 10 = 70
5 X 7 = 35
Ending Multiplication table of 7 thread.
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50
Ending Multiplication table of 5 thread.
Student@dblab-hp-04:~/Desktop/ooplab3$
```

Q2)

```
import java.util.*;

class Matrix implements Runnable{
    Thread t;
    int arr[];
    int sum;
    Matrix(int arr[]){
        t = new Thread(this, "row sum thread");
        this.arr = arr;
        sum = 0;
        t.start();
    }
    public void run(){
        for(int i = 0; i<arr.length; i++){
            sum+=arr[i];
        }
    }
}

public class l8q2 {
    public static void main (String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the dimensions of matrix : ");
        int row = sc.nextInt();
        int col = sc.nextInt();
        int[][] arr = new int[row][col];
        System.out.println("Enter the elements of the matrix : ");
        for(int i=0; i< row; i++){
            for(int j = 0; j< col; j++){
                arr[i][j] = sc.nextInt();
            }
        }
        Matrix[] a = new Matrix[row];
        for(int i=0; i<row; i++){
            a[i] = new Matrix(arr[i]);
        }
        for(int i=0; i<row; i++){
            try{
                a[i].t.join();
            }
            catch(Exception exc){
                System.out.println("Interrupted");
            }
        }
        int sum = 0;
        for(int i = 0; i< row; i++){
            sum += a[i].sum;
        }
        System.out.println("The sum is : " + sum);
    }
}
```

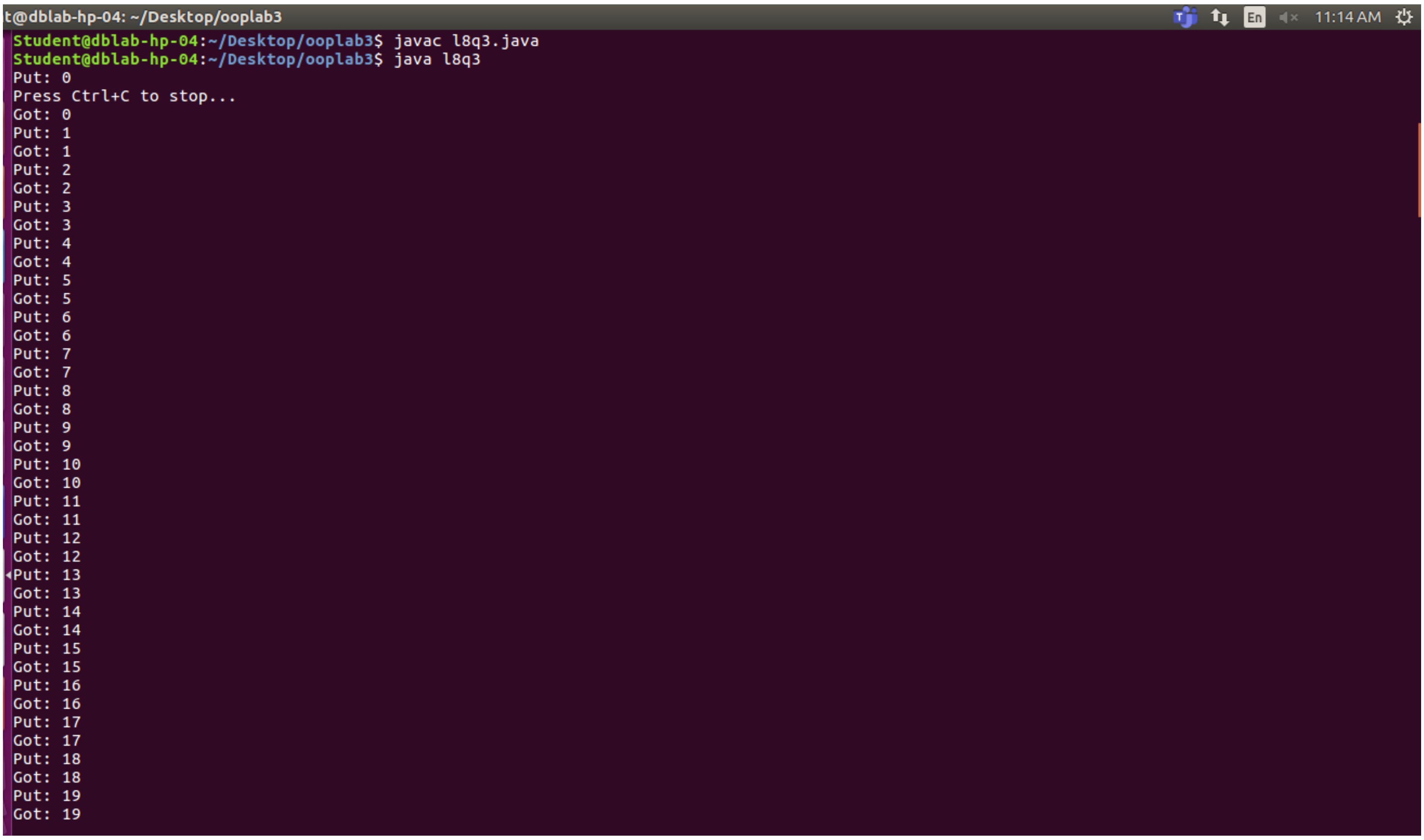


```
Student@dblab-hp-04:~/Desktop/ooplab3$ javac l8q2.java
Student@dblab-hp-04:~/Desktop/ooplab3$ java l8q2
Enter the dimensions of matrix :
3 3
Enter the elements of the matrix :
1 2 3 4 5 6 7 8 9
The sum is : 45
Student@dblab-hp-04:~/Desktop/ooplab3$ 
```

Q3)

```
class Q{
	int n;
	boolean valueSet = false;
	synchronized int get() {
		while(!valueSet)
			try {
				wait();
			}
			catch (InterruptedException e){
				System.out.println("InterruptedException caught");
			}
			System.out.println("Got: " + n);
			valueSet = false;
			notify();
			return n;
	}
	synchronized void put(int n) {
		while(valueSet)
			try {
				wait();
			}
			catch (InterruptedException e){
				System.out.println("InterruptedException caught");
			}
			this.n = n;
			valueSet = true;
```

```
            System.out.println("Put: " + n);
            notify();
    }
}



class Producer implements Runnable{
    Q q;
    Producer(Q q){
        this.q = q;
        new Thread(this, "Producer").start();
    }
    public void run() {
        int i=0;
        while(true) {
            q.put(i++);
        }
    }
}



class Consumer implements Runnable{
    Q q;
    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run() {
        while(true) {
            q.get();
        }
    }
}



class l8q3{
    public static void main(String[] args) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Ctrl+C to stop...");
    }
}
```



```
Student@dblab-hp-04:~/Desktop/ooplab3$ javac l8q3.java
Student@dblab-hp-04:~/Desktop/ooplab3$ java l8q3
Put: 0
Press Ctrl+C to stop...
Got: 0
Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3
Put: 4
Got: 4
Put: 5
Got: 5
Put: 6
Got: 6
Put: 7
Got: 7
Put: 8
Got: 8
Put: 9
Got: 9
Put: 10
Got: 10
Put: 11
Got: 11
Put: 12
Got: 12
Put: 13
Got: 13
Put: 14
Got: 14
Put: 15
Got: 15
Put: 16
Got: 16
Put: 17
Got: 17
Put: 18
Got: 18
Put: 19
Got: 19
```