**OOPL**

**Week 6**

Udeet Mittal

CSE C3

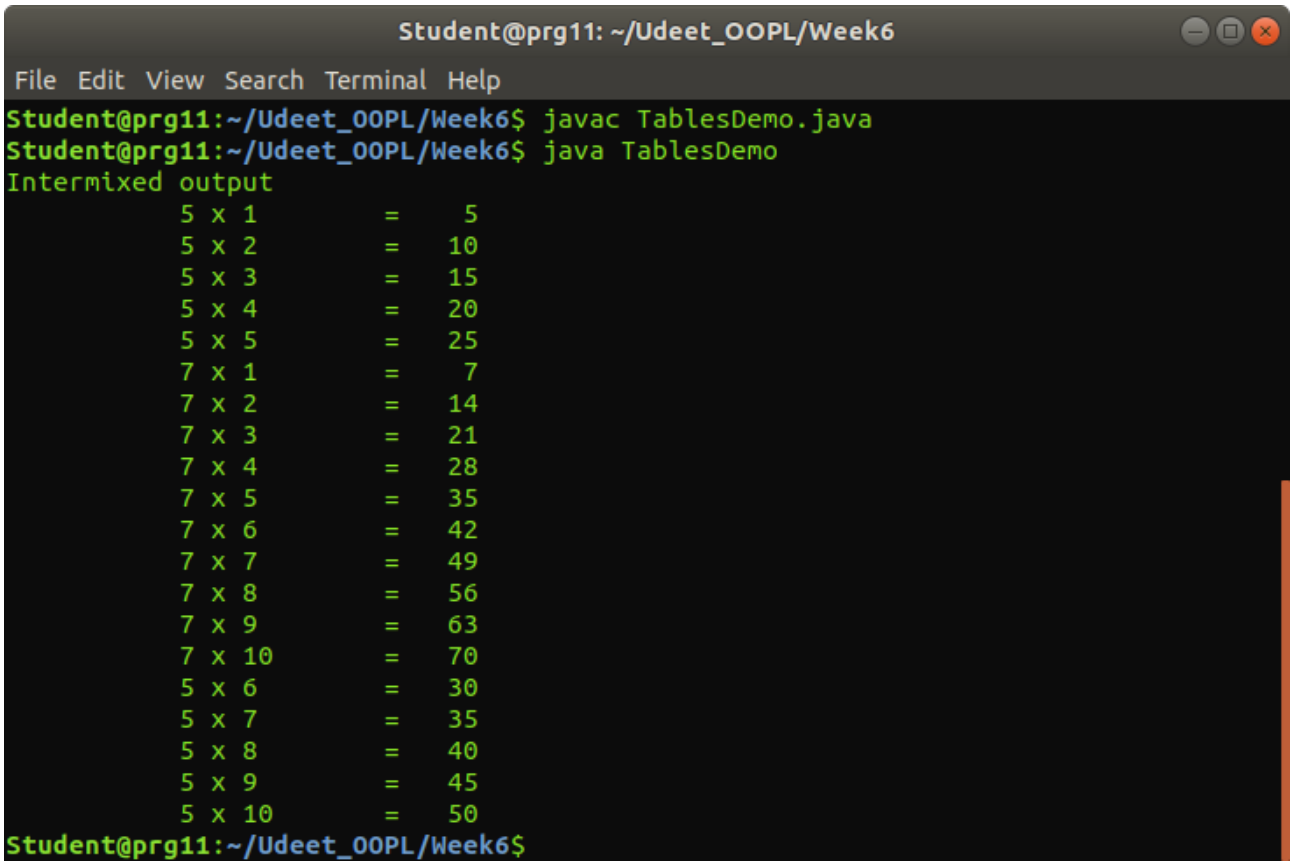
Roll Number 64

1.

**filename:TablesDemo.java (for intermixed output)**

```
import java.util.*;
class Table {
   void printTable(int n) throws InterruptedException {
      for (int i = 1; i <= 10; ++i) {
         System.out.printf("\t%4d x %d\t= %4d\n", n, i, n * i);
      }
   }
}
class MxNTableThread extends Thread {
   Table table;
   int n;
   MxNTableThread(Table t, int n) {
      table = t;
      this.n = n;
      start();
   }
   public void run() {

         try {

            table.printTable(n);
         } catch (InterruptedException e) {
            System.err.println("Thread Interrupted: " + e);
         }
   }
}
class TablesDemo{
   public static void main(String[] args) {
      Table t = new Table();
      System.out.println("Intermixed output");
      MxNTableThread t1 = new MxNTableThread(t, 5);
```

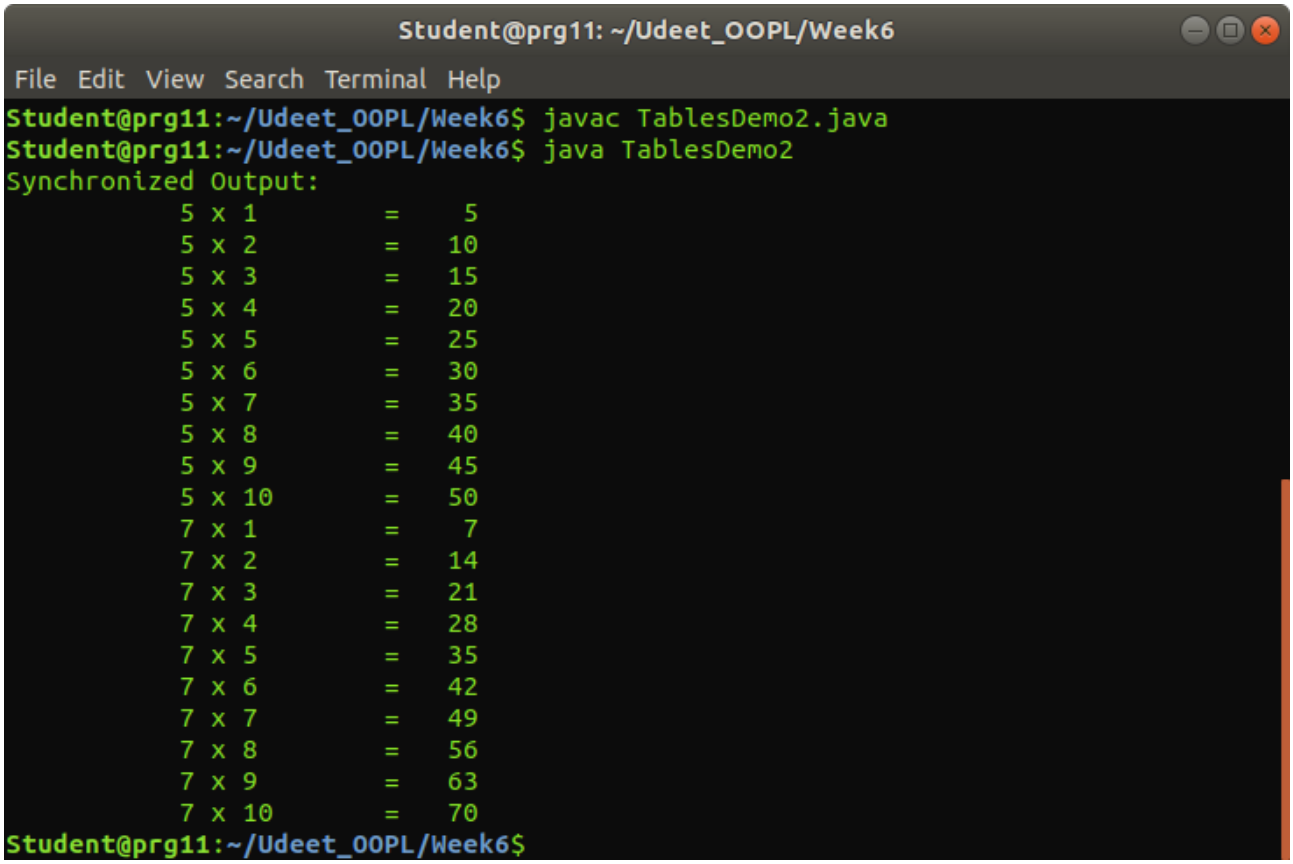
```
        MxNTableThread t2 = new MxNTableThread(t, 7);
    }
}
```




**filename:TablesDemo2.java (for synchronized output)**

```
import java.util.*;
class Table {
    void printTable(int n) throws InterruptedException {
        for (int i = 1; i <= 10; ++i) {
            System.out.printf("\t%4d x %d\t= %4d\n", n, i, n * i);
        }
    }
}
class MxNTableThread extends Thread {
    Table table;
    int n;
    MxNTableThread(Table t, int n) {
        table = t;
        this.n = n;
        start();
    }
    public void run() {
        synchronized(table) {
```

```
            try {
                table.printTable(n);
            } catch (InterruptedException e) {
                System.err.println("Thread Interrupted: " + e);
            }
        }
    }
}
class TablesDemo2{
    public static void main(String[] args) {
        Table t = new Table();
        System.out.println("Synchronized output");
        MxNTableThread t1 = new MxNTableThread(t, 5);
        MxNTableThread t2 = new MxNTableThread(t, 7);
    }
}
```

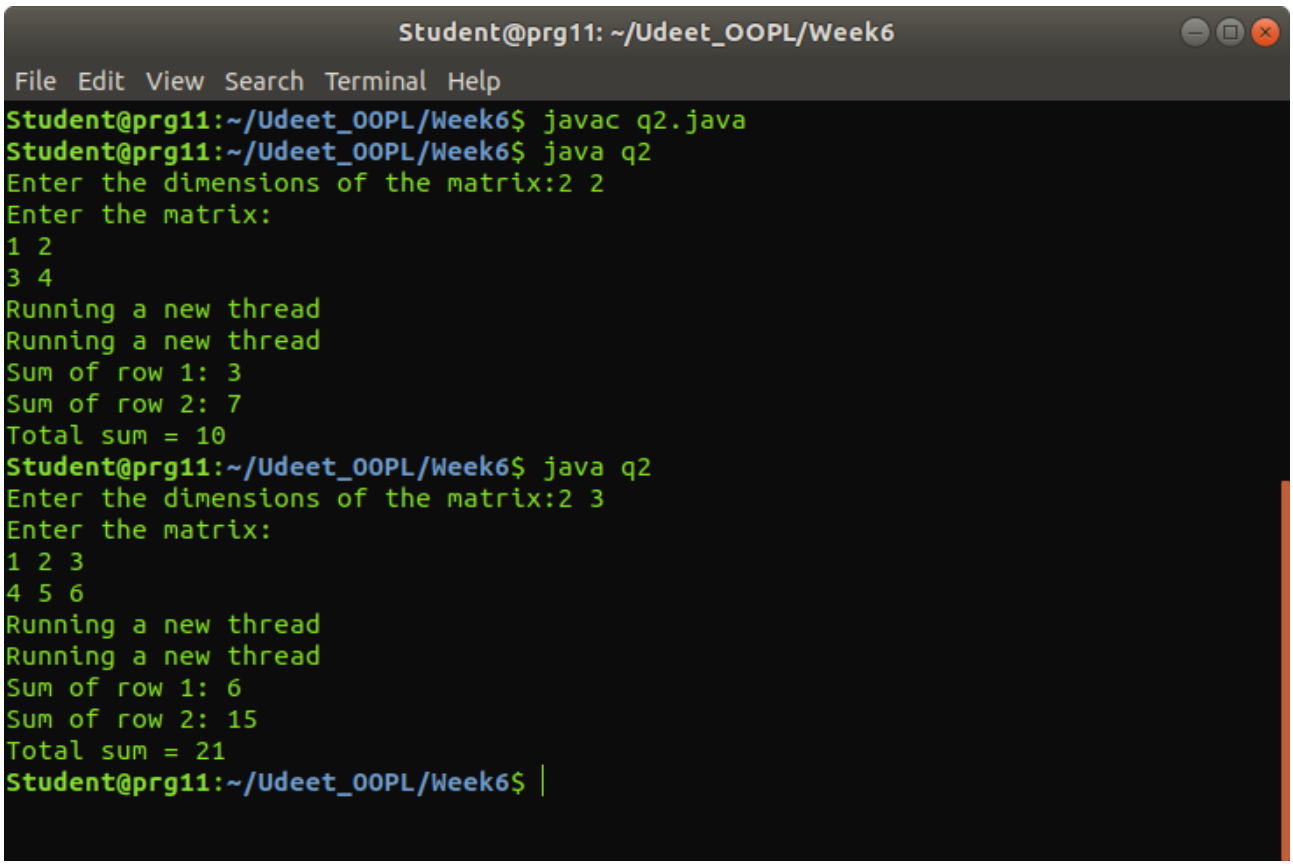


2.

**filename:q2.java**

```
import java.util.Scanner;
class Matrix{
	private int mat[][];

	public Matrix(int n, int m)
	{
		mat = new int[n][m];
	}

	public int[] getRow(int i)
	{
		return mat[i];
	}

	public void input(){
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the matrix:");
		for(int i=0; i<mat.length; i++)
		{
			for(int j=0; j<mat[i].length; j++)
				mat[i][j] = sc.nextInt();
		}
	}
}

class RowSum implements Runnable
{
	private int arr[];
	private int sum;

	RowSum(int a[])
	{
		arr = a;
		sum = 0;
	}

	public int getRowSum()
	{
		return sum;
	}

	public void run()
	{
```

```
                System.out.println("Running a new thread");
                for (int i : arr)
                        sum += i;
        }
}

class q2
{
        public static void main(String [] args)
        {
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter the dimensions of the matrix:");
                int n = sc.nextInt();
                int m = sc.nextInt();
                Matrix matrix = new Matrix(n,m);
                matrix.input();
                Thread threads[] = new Thread[n];
                RowSum rowsum[] = new RowSum[n];
                for(int i=0; i<n; i++)
                {
                        rowsum[i] = new RowSum(matrix.getRow(i));
                        threads[i] = new Thread(rowsum[i]);
                        threads[i].start();
                }
                int sum = 0;
                try
                {
                        for(int i=0;i<n; i++)
                        {
                                threads[i].join();
                                sum += rowsum[i].getRowSum();
                                System.out.println("Sum of row "+(i+1)+":
"+rowsum[i].getRowSum());
                        }
                }
                catch (InterruptedException e)
                {
                        e.printStackTrace();
                }
                System.out.println("Total sum = "+sum);
        }
}
```

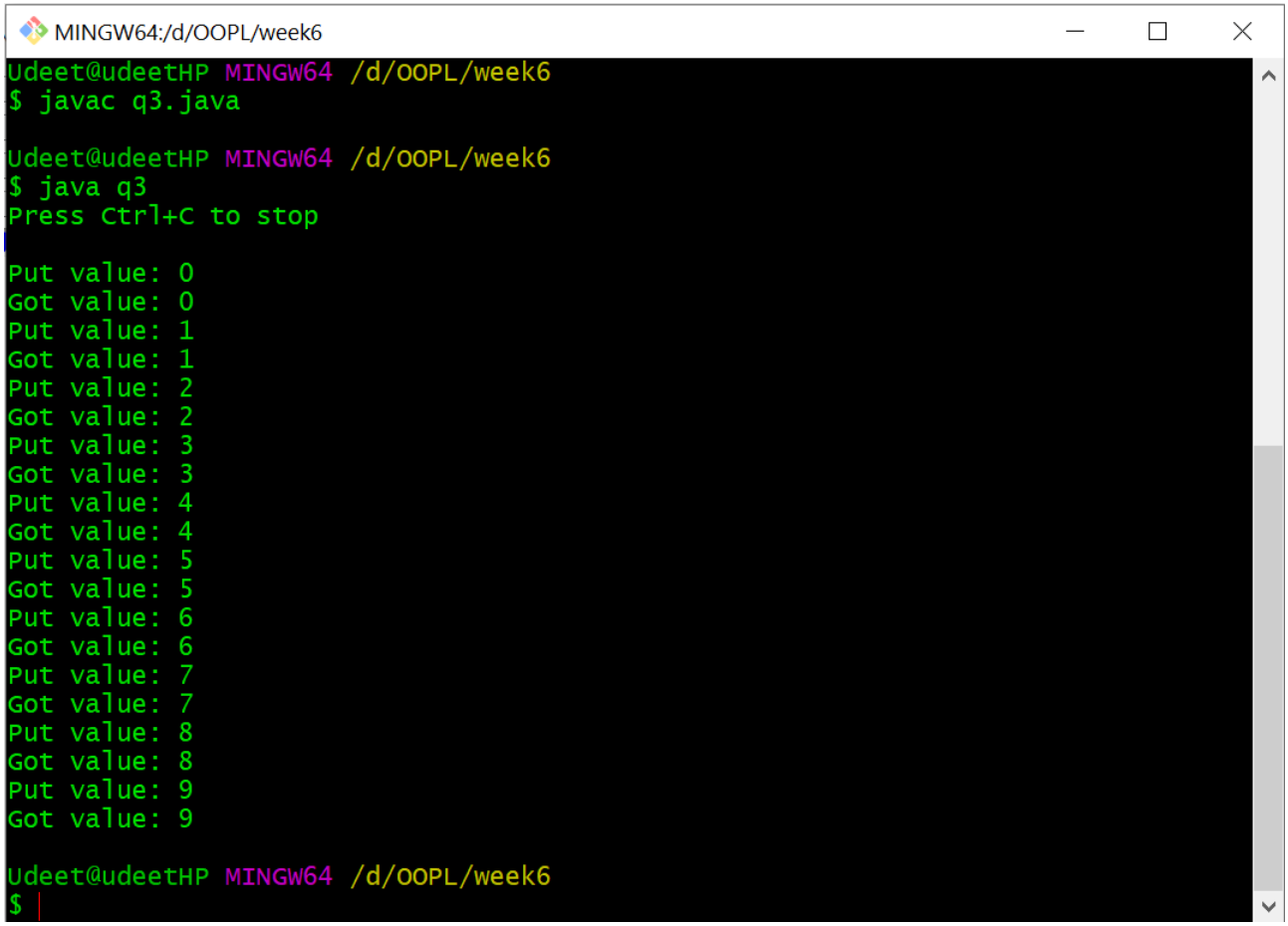



3.

**filename:q3.java**

```
import java.util.*;
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 value: " + 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 value: " + 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 (i < 10) {
            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 q3 {
    public static void main(String[] args) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Ctrl+C to stop\n");
    }
}
```


MINGW64:/d/OOPL/week6
Udeet@udeetHP MINGW64 /d/OOPL/week6
$ javac q3.java
Udeet@udeetHP MINGW64 /d/OOPL/week6
$ java q3
Press Ctrl+C to stop
Put value: 0
Got value: 0
Put value: 1
Got value: 1
Put value: 2
Got value: 2
Put value: 3
Got value: 3
Put value: 4
Got value: 4
Put value: 5
Got value: 5
Put value: 6
Got value: 6
Put value: 7
Got value: 7
Put value: 8
Got value: 8
Put value: 9
Got value: 9
Udeet@udeetHP MINGW64 /d/OOPL/week6
$