ASSIGNMENT 10 : LAB 9

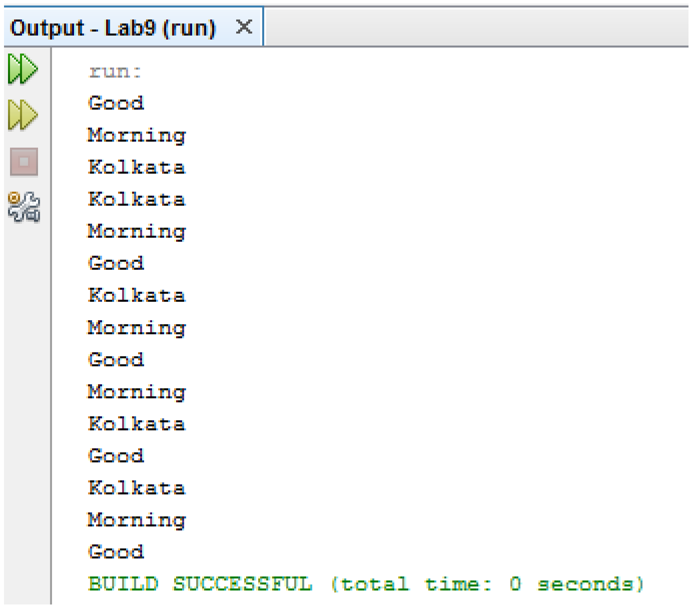
Q.1. Create a LoudHailer class that continually broadcasts a given woed and takes a deep breath for 1 second. Have 3 LoudHailer instances each running on a separate thread, to broadcast the words “Good”,”Morning”,”Kolkata” respectively. Observe the behavior of the 3 threads.

CODE:

```
/**
*
 * @author Harsh
 */
public class LoudHailer implements Runnable
{
   private Thread t;
   private String str;
   LoudHailer(String s)
   {
      str=s;
   }
   public void start()
   {
      if(t==null)
      {
         t=new Thread(this,str);//Thread initialized
         t.start();
      }
   }
   public void run()//Overridden
   {
      try
      {
         for(int i=1;i<=5;i++)
         {
            System.out.println(str);
            Thread.sleep(1000);//Thread waits
         }
      }
```

```
        catch(Exception e)
        {
           e.printStackTrace();
        }
    }
    public static void main(String[] args)
    {
        LoudHailer L1=new LoudHailer("Good");
        LoudHailer L2=new LoudHailer("Morning");
        LoudHailer L3=new LoudHailer("Kolkata");
        L1.start();
        L2.start();
        L3.start();
    }
}
```

OUTPUT:



```
run:
Good
Morning
Kolkata
Kolkata
Morning
Good
Kolkata
Morning
Good
Morning
Kolkata
Good
Kolkata
Morning
Good
BUILD SUCCESSFUL (total time: 0 seconds)
```

Question 1. Is the distribution of the words "Good","Morning" and "Kolkata" seemingly random?
Answer: Yes. The distribution is random. The threads are independent and hence they are displayed independently in a random fashion.

Question 2. What is the probability of having a "Good Morning Kolkata" broadcast in sequence?
Answer: The threads work independently. Since there are 3 words(threads), the total number of possible outcomes is 6. Hence its probability will be 1/6.

Question 3. What happens when you increase the priority of the thread that is broadcasting the word "Kolkata"?
Answer: On increasing the priority we observe that the number of times "Kolkata" gets the priority among the three threads is increased. However it isn't getting the priority all the times. Hence if we increase the priority, "Kolkata" is displayed at the top more number of times but not always.

Q.2. Using synchronization mechanisms, have the 3 threads from Question #1 co-operate, so that "Good Morning Kolkata" is broadcast in sequence every time.

CODE:

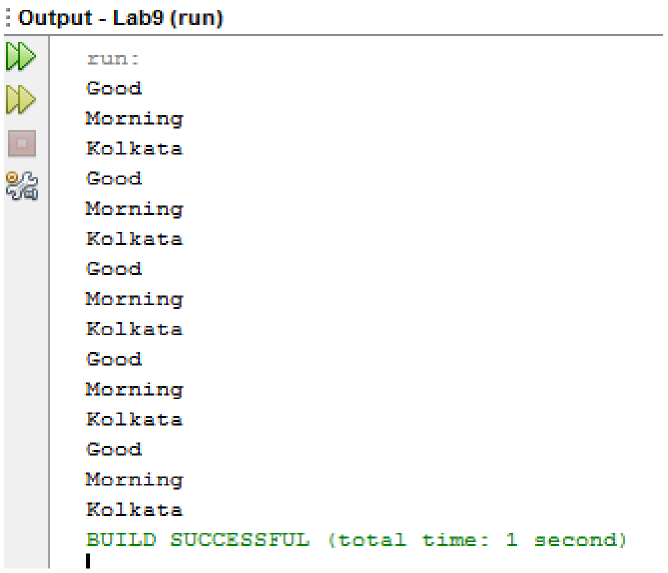
Sync.java:

```
/**
 *
 * @author Harsh
 */
public class Sync extends Thread
{
    private Thread t;
    private String str;
    Count ct=new Count();
    Sync(String s)
    {
str=s;
    }
    public void start()
    {
        if(t==null)
        {
            t=new Thread(this,str);//Thread initialized
t.start();
        }
   }
    public void run()
    {
        try
        {
            synchronized(ct)
            {
                for(int i=0;i<15;i++)
                {
                    if((this.ct.count()%3==0 &&str.compareTo("Good")==0 &&
i%3==0)||(this.ct.count()%3==1 &&
tr.compareTo("Morning")==0 && i%3==1)||(this.ct.count()%3==2
&&str.compareTo("Kolkata")==0 && i%3==2))
                    {//Checks for whose turn is it to display
System.out.println(str);
Thread.sleep(100);//Thread waits
                    }
ct.num++;
                }
            }
        }
        catch(Exception e)
        {
e.printStackTrace();
        }
    }
    public static void main(String[] args)
    {
        Sync S1=new Sync("Good");
        Sync S2=new Sync("Morning");
        Sync S3=new Sync("Kolkata");
        S1.start();
        S2.start();
        S3.start();
        try
        {
           S1.join();
            S2.join();
            S3.join();
        }
        catch(Exception e)
        {
e.printStackTrace();
        }
    }
}
```

Count.java:

```
public class Count
{
intnum=0;
   public int count()
   {
      return num;
   }
}
```

OUTPUT:



Output - Lab9 (run)

```
run:
Good
Morning
Kolkata
Good
Morning
Kolkata
Good
Morning
Kolkata
Good
Morning
Kolkata
Good
Morning
Kolkata
BUILD SUCCESSFUL (total time: 1 second)
```

Question 1. Is the use of the synchronized block or synchronized method sufficient to ensure a deterministic sequence?

Answer: No. A synchronized block is not sufficient. We need to have a method which will count for the number of occurrences of the thread and verify on that basis by checking the count and determining which thread's turn is it to be displayed.