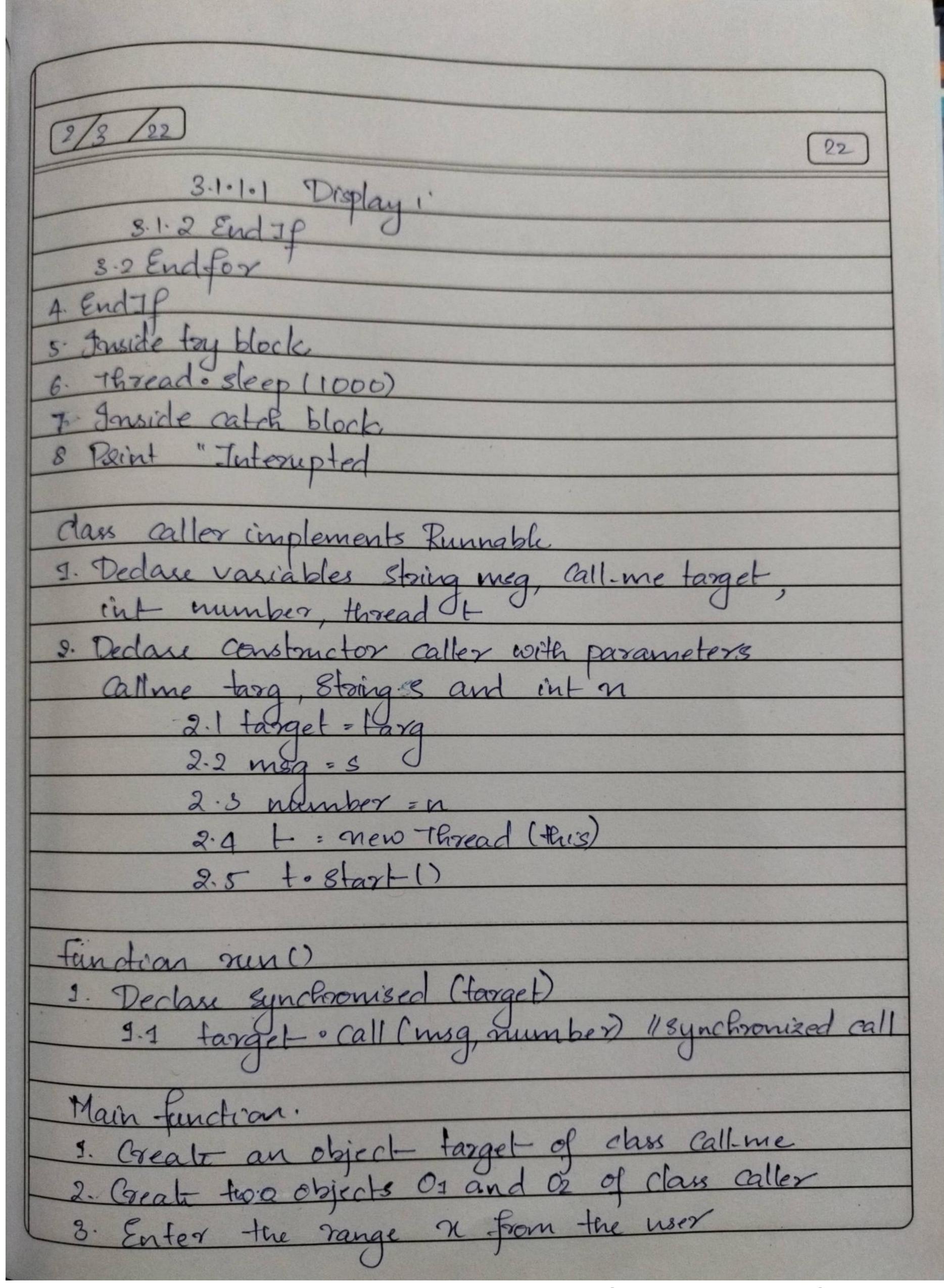
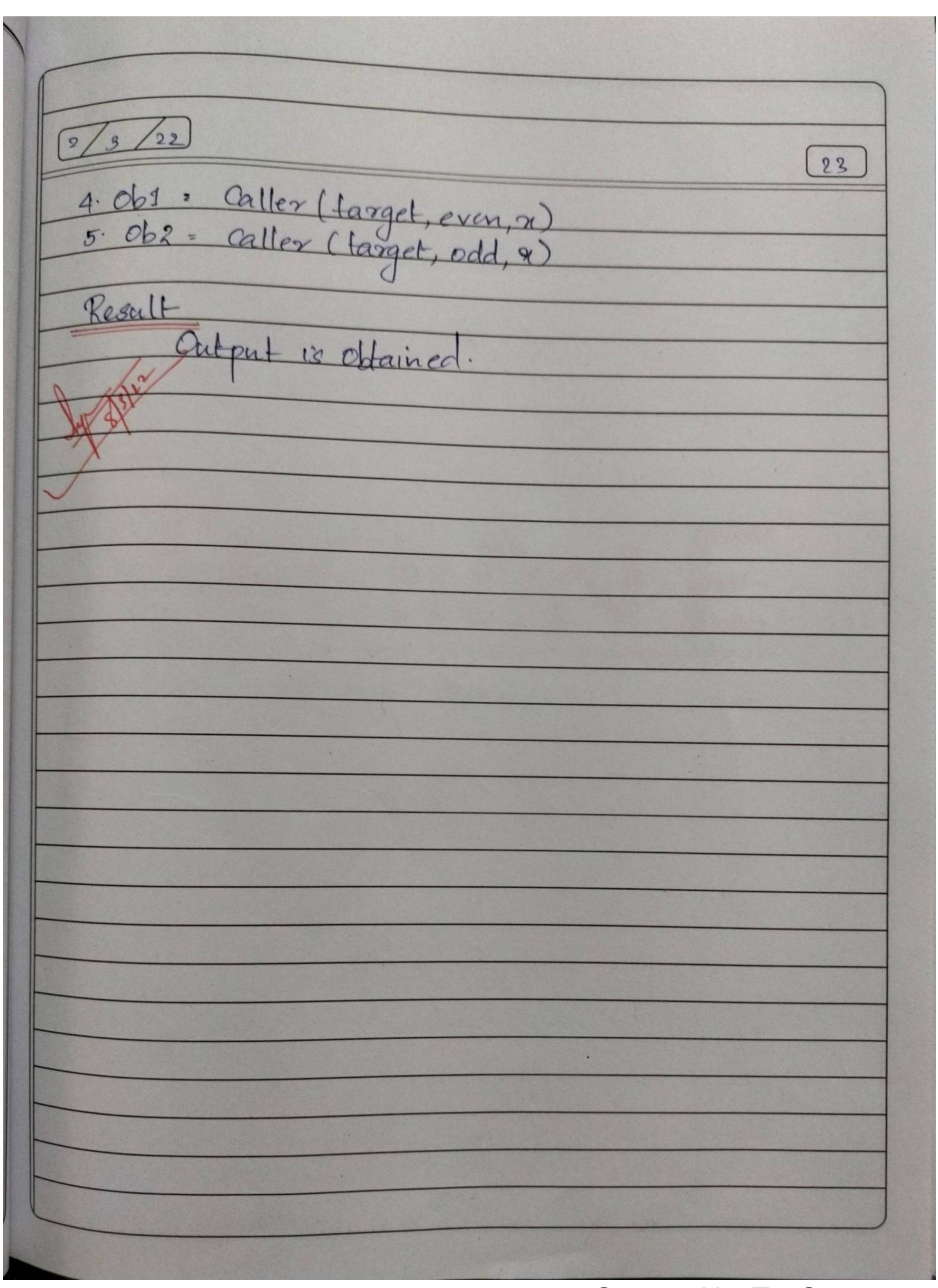
EXPERIMENT NO: 9 Thread Synchronication. Sunchronisation. Program that shows thread Inpu Enter the range. Expected output-The odd numbers and even numbers are painted seperately using thread synchronisation. Algorithm Class Call-me. 1. Declare function call with parameters string s and 2. IP 3 = "even" 2.1.1 4 10/0 2 = = 0 2.1.1.1 Display i 21.2 End If 2.2 Endfor 3. else





Output-Enter the limit: 15 LEven numbers are: 0 2 4 6 8 10 12 14) [Todd numbers are: 1 3 5 7 9 11 13 15]

Scanned by TapScanner

```
//Dovika B
//Section D : Qn 2
//Throad Synchronisation
import java.util. ";
class Callma (
    void call(int n) {
      System.out.print(n+" ");
      try
            Thread.sleep(1000);
      catch (InterruptedException e) {
            System.out.println("Interrupted");
   class Caller implements Runnable {
    String msg;
    Callme target;
    int number;
    Thread t:
    public Caller (Callmo targ, int n) {
    target = targ;
    number = n;
    t - new Thread (this);
    t.start();
    public void run() {
    synchronized(target) {
    target.call(number);
  class sync {
   public static void main (String args[]) [
       Scanner sc - new Scanner (System.in);
       int x;
       System.out.print("Enter the limit : ");
       x = sc.next.Tnt.();
     Callme target = new Callme();
      for (int i-0; i <-n; i++)
            if(i82--0)
           Caller ob1 - new Caller (target, i);
           Caller ob2 - new Caller (target, i);
   obl.t.join();
   ob2.t.join();
     catch (InterruptedEnception e) |
   System.out.println("Interrupted");
```