# EXPERIMENT 9

NAME= OM MITTAL

SAP ID = 500096091

ROLL NO= R2142210982

Write a program to implement the concept of threading by extending Thread Class and Runnable interface.

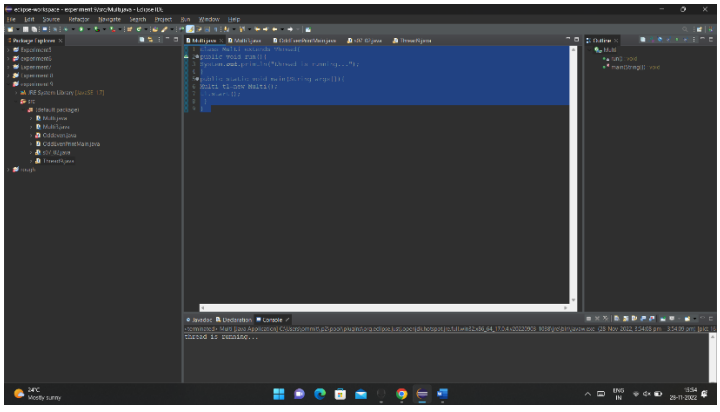
CODE:-

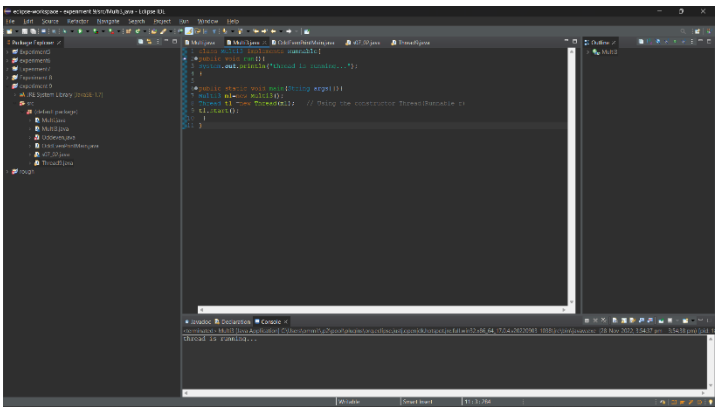
EXTENDING THREAD

```
class Multi extends Thread{
public void run(){
System.out.println("thread is running...");
}
public static void main(String args[]){
Multi t1=new Multi();
t1.start();
 }
}
```

EXTENDING RUNABBLE INTERFACE

```
class Multi3 implements Runnable{
public void run(){
System.out.println("thread is running...");
}

public static void main(String args[]){
Multi3 m1=new Multi3();
Thread t1 =new Thread(m1);   // Using the constructor Thread(Runnable r)
t1.start();
 }
}
```

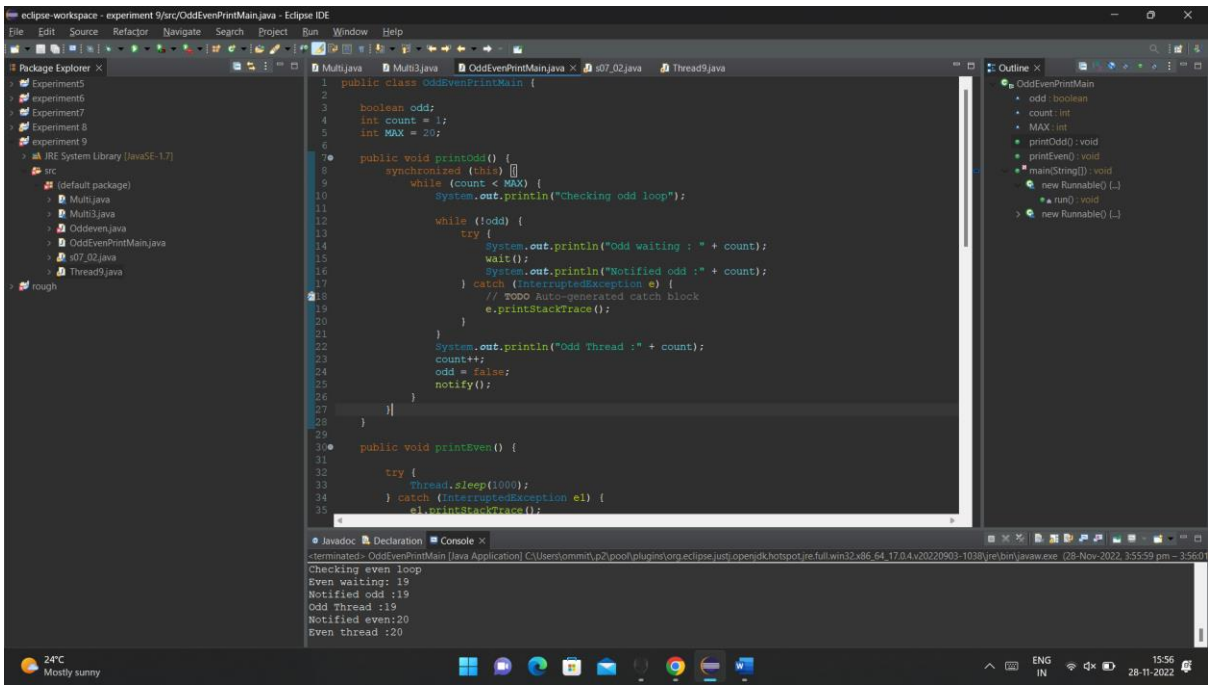




2) Write a program for generating 2 threads, one for printing even numbers and the other for printing odd numbers.

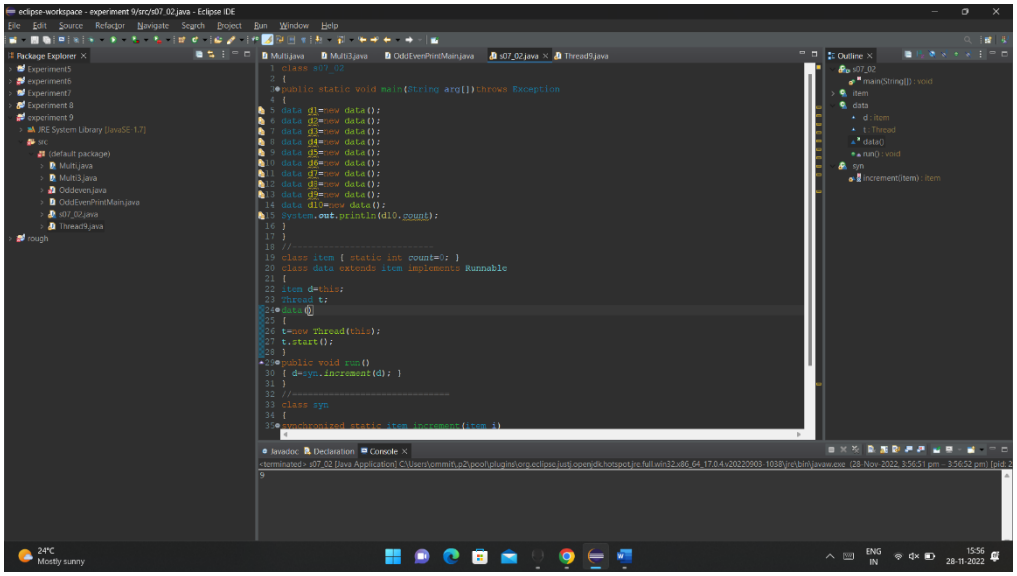
```
public class OddEvenPrintMain {

    boolean odd;
    int count = 1;
    int MAX = 20;

    public void printOdd() {
        synchronized (this) {
            while (count < MAX) {
                System.out.println("Checking odd loop");

                while (!odd) {
                    try {
                        System.out.println("Odd waiting : " + count);
                        wait();
                        System.out.println("Notified odd :" + count);
                    } catch (InterruptedException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }
                System.out.println("Odd Thread :" + count);
                count++;
                odd = false;
                notify();
            }
        }
    }

    public void printEven() {

        try {
            Thread.sleep(1000);
        } catch (InterruptedException e1) {
            e1.printStackTrace();
        }
        synchronized (this) {
            while (count < MAX) {
                System.out.println("Checking even loop");

                while (odd) {
                    try {
                        System.out.println("Even waiting: " + count);
                        wait();
                        System.out.println("Notified even:" + count);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                System.out.println("Even thread :" + count);
                count++;
                odd = true;
                notify();

            }
        }
    }

    public static void main(String[] args) {
```

```
        final OddEvenPrintMain oep = new OddEvenPrintMain();
        oep.odd = true;
        Thread t1 = new Thread(new Runnable() {

            @Override
            public void run() {
                oep.printEven();

            }
        });
        Thread t2 = new Thread(new Runnable() {

            @Override
            public void run() {
                oep.printOdd();

            }
        });

        t1.start();
        t2.start();

        try {
            t1.join();
            t2.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

    }
}
```



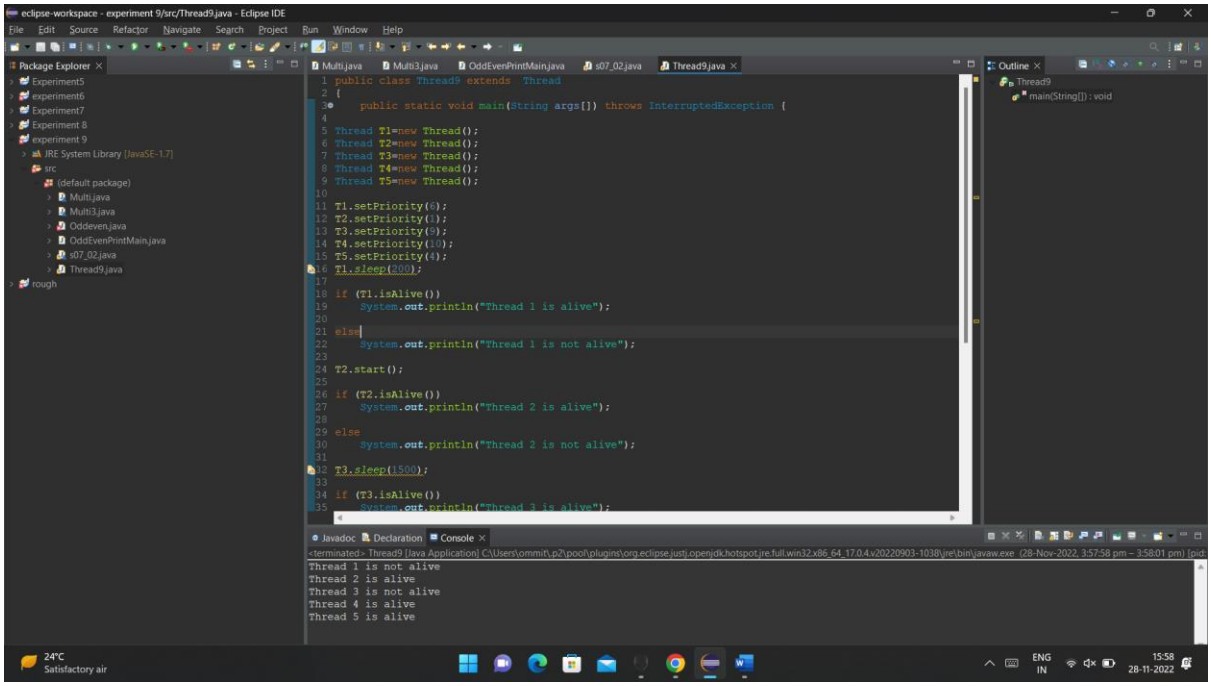
3) Write a program to launch 10 threads. Each thread increments a counter variable. Run the program with synchronization.

```
class s07_02
{
public static void main(String arg[])throws Exception
{
data d1=new data();
data d2=new data();
data d3=new data();
data d4=new data();
data d5=new data();
data d6=new data();
data d7=new data();
data d8=new data();
data d9=new data();
data d10=new data();
System.out.println(d10.count);
}
}
//----------------------------
class item { static int count=0; }
class data extends item implements Runnable
{
item d=this;
Thread t;
data()
{
t=new Thread(this);
t.start();
}
public void run()
{ d=syn.increment(d); }
}
//==============================
class syn
{
synchronized static item increment(item i)
{
i.count++;
return(i);
}
}
```



4)Write a Java program to create five threads with different priorities. Send two threads of the highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting

```
public class Thread9 extends  Thread
{
    public static void main(String args[]) throws InterruptedException {

Thread T1=new Thread();
Thread T2=new Thread();
Thread T3=new Thread();
Thread T4=new Thread();
Thread T5=new Thread();

T1.setPriority(6);
T2.setPriority(1);
T3.setPriority(9);
T4.setPriority(10);
T5.setPriority(4);
T1.sleep(200);

if (T1.isAlive())
    System.out.println("Thread 1 is alive");

else
    System.out.println("Thread 1 is not alive");

T2.start();

if (T2.isAlive())
    System.out.println("Thread 2 is alive");

else
    System.out.println("Thread 2 is not alive");

T3.sleep(1500);

if (T3.isAlive())
    System.out.println("Thread 3 is alive");

else
    System.out.println("Thread 3 is not alive");

T4.start();

if (T4.isAlive())
    System.out.println("Thread 4 is alive");

else
    System.out.println("Thread 4 is not alive");

T5.start();

if (T5.isAlive())
    System.out.println("Thread 5 is alive");

else
    System.out.println("Thread 5 is not alive");
    }
}
```


eclipse-workspace - experiment 9/src/Thread9.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Experiment5
experiment6
Experiment7
Experiment 8
experiment 9
JRE System Library [JavaSE-1.7]
src
(default package)
Multi.java
Multi3.java
Oddeven.java
OddEvenPrintMain.java
s07_02.java
Thread9.java
rough
Multi.java
Multi3.java
OddEvenPrintMain.java
s07_02.java
Thread9.java
1 public class Thread9 extends Thread
2 {
3 public static void main(String args[]) throws InterruptedException {
4
5 Thread T1=new Thread();
6 Thread T2=new Thread();
7 Thread T3=new Thread();
8 Thread T4=new Thread();
9 Thread T5=new Thread();
10
11 T1.setPriority(6);
12 T2.setPriority(1);
13 T3.setPriority(9);
14 T4.setPriority(10);
15 T5.setPriority(4);
16 T1.sleep(200);
17
18 if (T1.isAlive())
19 System.out.println("Thread 1 is alive");
20
21 else
22 System.out.println("Thread 1 is not alive");
23
24 T2.start();
25
26 if (T2.isAlive())
27 System.out.println("Thread 2 is alive");
28
29 else
30 System.out.println("Thread 2 is not alive");
31
32 T3.sleep(1500);
33
34 if (T3.isAlive())
35 System.out.println("Thread 3 is alive");
Outline
Thread9
main(String[]) : void
Javadoc Declaration Console
<terminated> Thread9 [Java Application] C:\Users\ommit\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903-1038\jre\bin\javaw.exe (28-Nov-2022, 3:57:58 pm – 3:58:01 pm) [pid
Thread 1 is not alive
Thread 2 is alive
Thread 3 is not alive
Thread 4 is alive
Thread 5 is alive
24°C
Satisfactory air
ENG
IN
15:58
28-11-2022