NAME:ROHAN NYATI

SAP ID:500075940

ROLL NO. : R177219148

BATCH-5(AI&ML)

EXPERIMENT NO – 9

TITLE: Threads

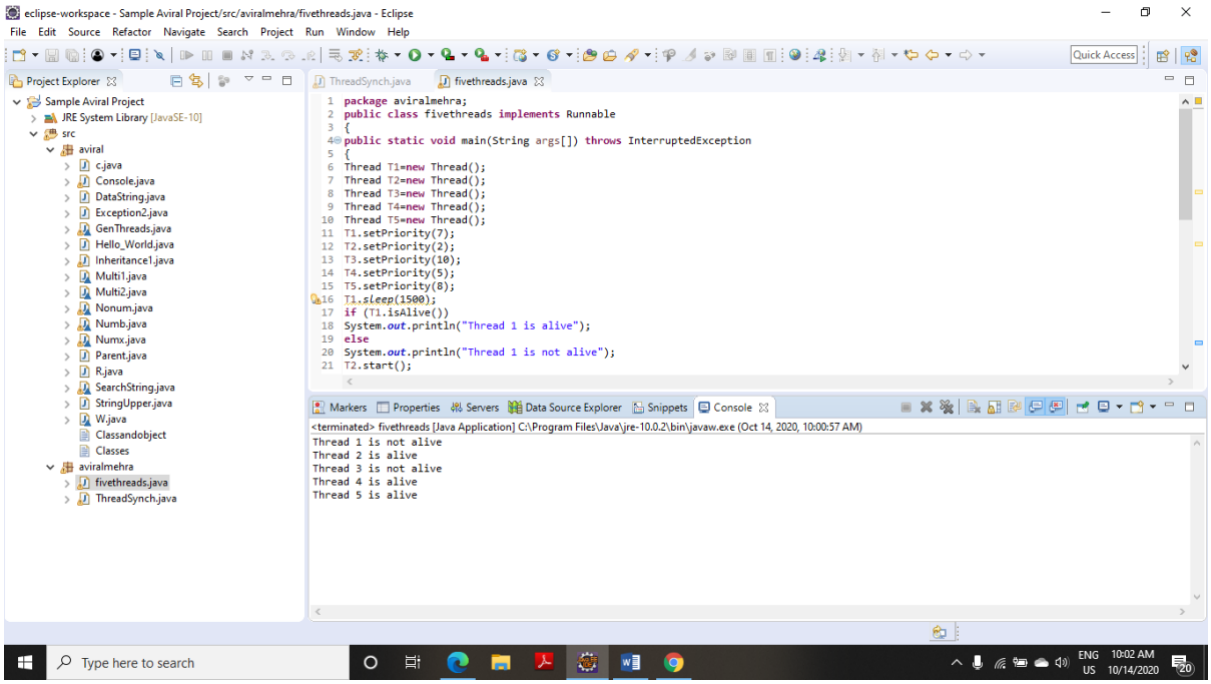
3. Write a Java program to create five threads with different priorities. Send two threads of the highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting.

CODE

```
package Rohan;
public class fivethreads implements Runnable
{
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(7);
T2.setPriority(2);
T3.setPriority(10);
T4.setPriority(5);
T5.setPriority(8);
T1.sleep(1500);
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(1000);
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");
}
}
```

OUTPUT



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

CODE

```
package Rohan;
class ThreadSynch
{
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);
  }
}
```

OUTPUT