Name: S Skanda

OOJ LAB TEST-2

USN: 1BM19CS137 Sem: 3 Section: C Date: 23/12/20

Batch: C2

4) Create a class as shown

```
class Counter{
    int count;
    void inc(){
        count =count+1;
      }
    int getCount(){
        return count;
     }
 }
```

Create 3 threads ~~with~~ that will call the inc() method on the same counter object. Start them all, and wait for all threads to terminate. Assign different priority to threads. Justify your output

```
class Counter extends Runnable{
    int count; ~~th~~ Thread t;
    Counter(int x, int p)
     { count = x;
       t=new Thread (this); t.setPriority(p); }
    void inc (){
        count =count+1;
      }
    int getCount(){
        return count;
      }
  public void run (){
        while (running){
            inc();
        }} ~~System.out.println(this.getCount());~~
    }
  public void stop(){
        running = false;
      }
```

```
public void start (){
        t.start ();
      }

}
class main_class {
 public static void main ( String args[])
{   Thread.currentThread(). setPriority(Thread.MAX_PRIORITY);
   ~~Counter C~~
   Counter C1 = new Counter ( 5, Thread.NORM_PRIORITY+1);
   Counter C2 = new Counter ( 8, Thread.NORM_PRIORITY+2);
   Counter C3 = new Counter ( 7, Thread.NORM_PRIORITY+3);

   ~~Count~~
    C1.start();
    C2.start();
    C3.start();
   try { Thread. sleep( 1000 )
        }
   catch ( InterruptedException e ){
       System.out.println("Main thread interrupted");
    };
    C1. stop ();
    C2. stop ();
    C3. stop ();
    ~~try~~{
     ~~C1.~~
    System.out.println ("Low priority thread"+ C1.getCount()
   System.out.println ("Medium priority thread"+ C2.getCount()}
   System.out.println ("High priority thread"+C3.getCount()
Try {
 C1.t.join();
 C2.t.join();
 C3.t.join(); }
```

```
  Catch (InterruptedException e){
   System.out.println("Interrupt exception caught"); }
  }
}
```