Name-Pradnya.Abhay.Magennavar.
Roll no-12
Experiment no-13
Experiment name-Implementing java programs based on multithreading.

Multithreading in Java

Multithreading in Java is a process of executing multiple threads simultaneously.
A thread is a lightweight sub-process, the smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.
However, we use multithreading than multiprocessing because threads use a shared memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.

What is Thread in java

A thread is a lightweight subprocess, the smallest unit of processing. It is a separate path of execution.
Threads are independent. If there occurs exception in one thread, it doesn't affect other threads. It uses a shared memory area.

1.program on multithreading using thread class.

```
class my extends Thread
{
  public void run()
        {
                try{
                        System.out.println("Thread:"+Thread.currentThread().getId());
                        Thread.sleep(1000);
                 }
                 catch(Exception e)
                {
                        System.out.println(e);
                }
        }
}
class me
{
        public static void main(String args[])
        {
                int n=5;

                for(int i=1;i<5;i++)
                {
                        my m1=new my();
                        m1.start();
                        System.out.println(m1.getState());
                }
```

```
        }
}
```

Output-

```
C:\Users\LENOVO-PC\Desktop>java me
RUNNABLE
RUNNABLE
RUNNABLE
RUNNABLE
Thread:22
Thread:20
Thread:23
Thread:21
```

2.program for multithreading using runnable interface.

```
class my1 implements Runnable
{
        public void run()
        {
                try{
                        System.out.println("Thread:"+Thread.currentThread().getId());
                        Thread.sleep(1000);
                  }
                catch(Exception e)
                {
                        System.out.println(e);
                }
        }
}
class hari
{
        public static void main(String args[])
        {
                int n=10;
                for(int i=1;i<10;i++)
                {
                        Thread t1=new Thread(new my1());
                        t1.start();
                        System.out.println(t1.getState());
                }
        }
}
```

Output-

```
Note: me2.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\Vaishnavi\Desktop\v>javac -Xlint:deprecation me2.java
me2.java:6: warning: [deprecation] getId() in Thread has been deprecated
                      System.out.println("Thread:"+Thread.currentThread().getId());
                                                                           ^
1 warning

C:\Users\Vaishnavi\Desktop\v>java me2
RUNNABLE
RUNNABLE
RUNNABLE
RUNNABLE
RUNNABLE
RUNNABLE
RUNNABLE
RUNNABLE
RUNNABLE
Thread:22
Thread:23
Thread:21
Thread:24
Thread:28
Thread:29
Thread:26
Thread:25
Thread:27
```