OOPS LAB ACTIVITY - 9

Name: Sarvagya Gupta
Roll No. : R2142201047
Sap Id : 500083195

Tittle :- Threads

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

Code

```java


public class thread1 extends Thread
{
    public  void run()
    {
        System.out.println("thread is running");//creating thread by extending thread class
    }
    Run | Debug
    public static void main(String args[])
    {
        thread1 t = new thread1();
        t.start();
    }
}


```

Output

```
PS E:\codes\java> cd "e:\codes\java\lab 9\q1\" ; if ($?) { javac thread1.java } ; if ($?) { java thread1 }
thread is running
PS E:\codes\java\lab 9\q1>
```

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

Code

```
public class even extends Thread
{
    public void run()
    {
        for(int i=0 ; i<10 ; i++)
        {
            if (i%2 == 0)
            System.out.println(i + "even");
        }
    }
}


public class odd extends Thread
{
    public void run()
    {
        for (int i=0 ; i<10 ; i++)
        {
            if(i%2 != 0)
            System.out.println(i + "odd");
        }
    }
}

^

public class thread7
{
    Run | Debug
    public static void main (String args [])
    {
        even e = new even();
        odd o = new odd();
        e.start();
        o.start();
    }
}
```

Output

```
PS E:\codes\java> cd "e:\codes\java\lab 9\q2\" ; if ($?) { javac thread7.java } ; if ($?) { java thread7 }
0even
1odd
2even
3odd
4even
5odd
6even
7odd
8even
9odd
PS E:\codes\java\lab 9\q2>
```

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

Code

```java

public class Counter
{
    static Thread[] threads = new Thread[10];
    Run | Debug
    public static void main (String args [])
    {
        Count c = new Count();
        for (int i=0 ; i<10 ; i++)
        {
            threads[i] = new Thread(c);
            threads[i].start();
        }
    }
}

class Count implements Runnable
{
    int n = 1;

    @Override
    public void run()
    {
        System.out.println(n++);
    }

    public void showOutput()
    {
        System.out.println(n++);
    }
}
```

Output

```
1
2
3
4
5
6
7
8
9
10
```

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

Code

```
class thread9 extends  Thread
{
    Run | Debug
    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");

```

```
37    else
38        System.out.println("Thread 3 is not alive");
39
40    T4.start();
41
42    if (T4.isAlive())
43        System.out.println("Thread 4 is alive");
44
45    else
46        System.out.println("Thread 4 is not alive");
47
48    T5.start();
49
50    if (T5.isAlive())
51        System.out.println("Thread 5 is alive");
52
53    else
54        System.out.println("Thread 5 is not alive");
55        }
56    }
57
58
59
```

Output

```
PS E:\codes\java> cd "e:\codes\java\lab 9\q4\" ; if ($?) { javac thread9.java } ; if ($?) { java thread9 }
Thread 1 is not alive
Thread 2 is alive
Thread 3 is not alive
Thread 4 is alive
Thread 5 is alive
PS E:\codes\java\lab 9\q4>
```