

## **EXPERIMENT – 10**

**TITLE:** Threads

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

```
class ImplementsRunnable implements Runnable {

    private int counter = 0;

    public void run() {
        counter++;
        System.out.println("ImplementsRunnable : Counter : " + counter);
    }
}

class ExtendsThread extends Thread {

    private int counter = 0;

    public void run() {
        counter++;
        System.out.println("ExtendsThread : Counter : " + counter);
    }
}

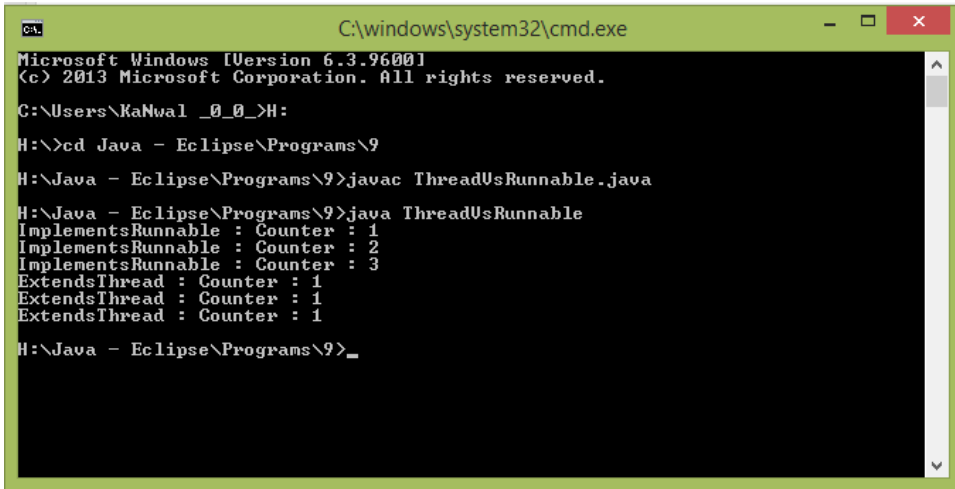
public class ThreadVsRunnable {

    public static void main(String args[]) throws Exception {
        //Multiple threads share the same object.
        ImplementsRunnable rc = new ImplementsRunnable();
        Thread t1 = new Thread(rc);
        t1.start();
        Thread.sleep(1000); // Waiting for 1 second before starting next thread
        Thread t2 = new Thread(rc);
        t2.start();
        Thread.sleep(1000); // Waiting for 1 second before starting next thread
        Thread t3 = new Thread(rc);
        t3.start();

        //Creating new instance for every thread access.
        ExtendsThread tc1 = new ExtendsThread();
        tc1.start();
        Thread.sleep(1000); // Waiting for 1 second before starting next thread
        ExtendsThread tc2 = new ExtendsThread();
        tc2.start();
        Thread.sleep(1000); // Waiting for 1 second before starting next thread
        ExtendsThread tc3 = new ExtendsThread();
        tc3.start();
    }
}
```

```
}
```

## Output :



```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\KaNwal _0_0_>H:
H:\>cd Java - Eclipse\Programs\9
H:\Java - Eclipse\Programs\9>javac ThreadUsRunnable.java
H:\Java - Eclipse\Programs\9>java ThreadUsRunnable
ImplementsRunnable : Counter : 1
ImplementsRunnable : Counter : 2
ImplementsRunnable : Counter : 3
ExtendsThread : Counter : 1
ExtendsThread : Counter : 1
ExtendsThread : Counter : 1
H:\Java - Eclipse\Programs\9>_
```

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

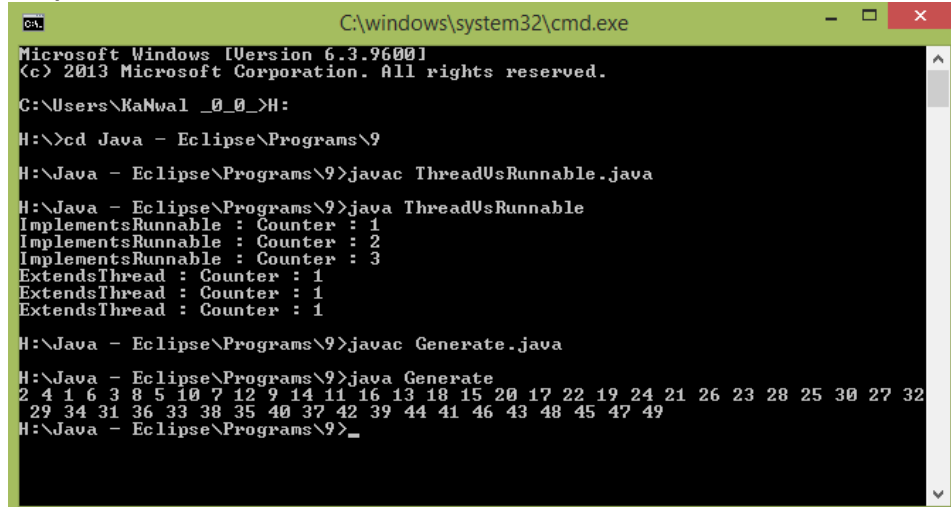
```
class even extends Thread
{
Thread t=null; even()
{
t=new Thread(this); start();
}
public void run()
{
try
{
for(int i=2;i<50;i+=2) System.out.print(i+" "); Thread.sleep(100);
}
catch(Exception e)
{System.out.println("thread interepted");}
}
}
class odd extends Thread
{
Thread t=null; odd()
{
t=new Thread(this); start();
}
public void run()
{
try
{
for(int i=1;i<50;i+=2) System.out.print(i+" "); Thread.sleep(100);
}
catch(Exception e)
{System.out.println("thread interepted");}
}
}
class Generate
{
```

```

public static void main(String arg[])
{
even e=new even(); odd o=new odd();
}
}

```

#### Output :



```

C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\KaNwal _0_0_>H:
H:\>cd Java - Eclipse\Programs\9
H:\Java - Eclipse\Programs\9>javac ThreadUsRunnable.java
H:\Java - Eclipse\Programs\9>java ThreadUsRunnable
ImplementsRunnable : Counter : 1
ImplementsRunnable : Counter : 2
ImplementsRunnable : Counter : 3
ExtendsThread : Counter : 1
ExtendsThread : Counter : 1
ExtendsThread : Counter : 1
H:\Java - Eclipse\Programs\9>javac Generate.java
H:\Java - Eclipse\Programs\9>java Generate
2 4 1 6 3 8 5 10 7 12 9 14 11 16 13 18 15 20 17 22 19 24 21 26 23 28 25 30 27 32
29 34 31 36 33 38 35 40 37 42 39 44 41 46 43 48 45 47 49
H:\Java - Eclipse\Programs\9>_

```

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

```

class number
{
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 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");  
}  
}
```