

```

1  -----CounterThread.java-----
2  package com.ameya.threads;
3
4  public class CounterThread extends Thread {
5      public CounterThread(String name,int priority) {
6          this.setName(name);
7          //Priorities=1-10, Where 1=MIN_PRIORITY,
8          //5=NORM_PRIORITY, 10=MAX_PRIORITY
9          this.setPriority(priority);
10         System.out.println(this);
11     }
12     public void run() {
13         for(int cnt=1;cnt<=10;cnt++) {
14             System.out.println(this.getName()+" => "+cnt);
15             try {
16                 Thread.sleep(1000);
17             } catch (InterruptedException e) {
18                 e.printStackTrace();
19             }
20         }
21         System.out.println(this.getName()+" TERMINATED.... !!!");
22     }
23  }
24  -----TestExtendsThread.java-----
25  package com.ameya.test;
26
27  import com.ameya.threads.CounterThread;
28
29  public class TestExtendsThread {
30      public static void main(String[] args) {
31          CounterThread cThread1=new CounterThread("Thread-1",
32              Thread.NORM_PRIORITY);
33          CounterThread cThread2=new CounterThread("Thread-2",
34              Thread.NORM_PRIORITY+1);
35          CounterThread cThread3=new CounterThread("Thread-3",
36              Thread.NORM_PRIORITY+3);
37          cThread1.start();
38          cThread2.start();
39          cThread3.start();
40          System.out.println(Thread.currentThread());

```

```

38     for(int cnt=1;cnt<=5;cnt++) {
39         System.out.println("MainThread => "+cnt);
40         try {
41             Thread.sleep(1000);
42         } catch (InterruptedException e) {
43             e.printStackTrace();
44         }
45     }
46     printThreadStatus(cThread1, cThread2, cThread3);
47     try {
48         cThread1.join();
49         cThread3.join();
50         cThread2.join();
51     } catch (InterruptedException e) {
52         e.printStackTrace();
53     }
54     printThreadStatus(cThread1, cThread2, cThread3);
55     System.out.println("MainThread TERMINATED...!!!");
56 }
57 static void printThreadStatus(CounterThread cThread1,
58     CounterThread cThread2,
59     CounterThread cThread3) {
60     System.out.println("Thread-1 is Alive => "+cThread1.isAlive());
61     System.out.println("Thread-2 is Alive => "+cThread2.isAlive());
62     System.out.println("Thread-3 is Alive => "+cThread3.isAlive());
63 }
64 }
65 -----CounterThreadRunnable.java-----
66 package com.ameya.threads;
67
68 public class CounterThreadRunnable implements Runnable {
69
70     private Thread thread;
71     public CounterThreadRunnable(String name,int priority) {
72         thread=new Thread(this);
73         thread.setName(name);
74         thread.setPriority(priority);
75         System.out.println(thread);
76     }
77     @Override
78     public void run() {

```

```

79         for(int cnt=1;cnt<=10;cnt++) {
80             System.out.println(thread.getName()+" => "+cnt);
81             try {
82                 Thread.sleep(1000);
83             } catch (InterruptedException e) {
84                 e.printStackTrace();
85             }
86         }
87         System.out.println(thread.getName()+" TERMINATED.... !!!");
88     }
89     public Thread getThread() {
90         return thread;
91     }
92 }
93 -----TestRunnableThread.java-----
94 package com.ameya.test;
95
96 import com.ameya.threads.CounterThreadRunnable;
97
98 public class TestRunnableThread {
99
100     public static void main(String[] args) {
101         CounterThreadRunnable cThread1=new
102         CounterThreadRunnable("Thread-1", 5);
103         CounterThreadRunnable cThread2=new
104         CounterThreadRunnable("Thread-2", 7);
105         CounterThreadRunnable cThread3=new
106         CounterThreadRunnable("Thread-3", 8);
107         cThread1.getThread().start();
108         cThread2.getThread().start();
109         cThread3.getThread().start();
110         System.out.println(Thread.currentThread());
111         for(int cnt=1;cnt<=5;cnt++) {
112             System.out.println("MainThread => "+cnt);
113             try {
114                 Thread.sleep(1000);
115             } catch (InterruptedException e) {
116                 e.printStackTrace();

```

```
117      "+cThread1.getThread().isAlive());
      System.out.println("cThread2 isAlive =>
118      "+cThread2.getThread().isAlive());
      System.out.println("cThread3 isAlive =>
      "+cThread3.getThread().isAlive());
119      try {
120          cThread1.getThread().join();
121          cThread2.getThread().join();
122          cThread3.getThread().join();
123      } catch (InterruptedException e) {
124          e.printStackTrace();
125      }
126      System.out.println("cThread1 isAlive =>
      "+cThread1.getThread().isAlive());
127      System.out.println("cThread2 isAlive =>
      "+cThread2.getThread().isAlive());
128      System.out.println("cThread3 isAlive =>
      "+cThread3.getThread().isAlive());
129      System.out.println("MainThread TERMINATED...!!!");
130  }
131
132 }
133
```