Multithreading

Thread Control Mechanism

Sensitivity: Internal & Restricted

Agenda

1 **Thread Control Mechanism**

Sensitivity: Internal & Restricted

Objectives

At the end of this module, you will be able to:

- Thread Control Mechanism

Thread Control Mechanism

Sensitivity: Internal & Restricted

Control Thread Execution

- Two ways exist by which you can determine whether a thread has finished:

- The **isAlive( )** method will return true if the thread upon which it is called is still running; else it will return false

- The **join( )** method waits until the thread on which it is called terminates.
- Syntax:
- final boolean isAlive()
- final void join() throws InterruptedException

Control Thread Execution (Contd.).

```
public class DemoThread implements Runnable {
String name;
Thread thread;
DemoThread(String threadname) {
name = threadname;
thread = new Thread(this, name);
System.out.println("New Thread: " + thread);
thread.start();
}
```

Control Thread Execution (Contd.).

```
public void run() {
try {
        for(int i=5; i>0; i--) {
System.out.println("Child Thread: " + i);
Thread.sleep(1000); }
       }
catch (InterruptedException e) {
          System.out.println(name + "Interrupted");
}
System.out.println(name +"Exiting");
}}
```

Sensitivity: Internal & Restricted

Control Thread Execution (Contd.).

```
public class MultiThreadImpl {
public static void main(String args[]) {
DemoThread t1 = new DemoThread("One");
DemoThread t2 = new DemoThread("Two");
DemoThread t3 = new DemoThread("Three");
System.out.println("Thread One is alive: " +
t1.thread.isAlive());
System.out.println("Thread Two is alive: " +
t2.thread.isAlive());
System.out.println("Thread Three is alive: "  +
t1.thread.isAlive());
```

Control Thread Execution (Contd.).

```
        try {
       System.out.println("Waiting for child  threads to
finish");
       t1.thread.join();
       t2.thread.join();
       t3.thread.join();
}
catch (InterruptedException e) {
       System.out.println("Main thread interrupted");
}
```

Control Thread Execution (Contd.).

```
System.out.println("Thread One is alive: " +
t1.thread.isAlive());

System.out.println("Thread One is alive: " +
t1.thread.isAlive());

System.out.println("Thread One is alive: " +
t1.thread.isAlive());

System.out.println("Main thread exiting");

}

}
```

Control Thread Execution (Contd.).

Output:

```
New Thread: Thread[One,5,main]
New Thread: Thread[Two,5,main]
Child Thread: 5
New Thread: Thread[Three,5,main]
Thread One is alive: true
Thread Two is alive: true
Thread Three is alive: true
Waiting for child  threads to finish
Child Thread: 5
Child Thread: 5
Child Thread: 4
Child Thread: 4
Child Thread: 4
```

Continued….

```
Child Thread: 3
Child Thread: 3
Child Thread: 3
Child Thread: 2
Child Thread: 2
Child Thread: 2
Child Thread: 1
Child Thread: 1
Child Thread: 1
ThreeExiting
OneExiting
TwoExiting
Thread One is alive: false
Thread One is alive: false
Thread One is alive: false
Main thread exiting
```

Assignment

Sensitivity: Internal & Restricted

Summary

- Different Thread control mechanisms

Thank You

Sensitivity: Internal & Restricted