

Java Threads - 2

Joining

- One thread can wait (with or without a timeout) for another thread (the target) to terminate by issuing the **join** method call on the target's thread object
- The **isAlive** method allows a thread to determine if the target thread has terminated

Thread Example without Join

```
public class MyThread implements Runnable {  
    public void run() {  
        System.out.println(" this thread is now exiting ... ");  
    }  
}
```

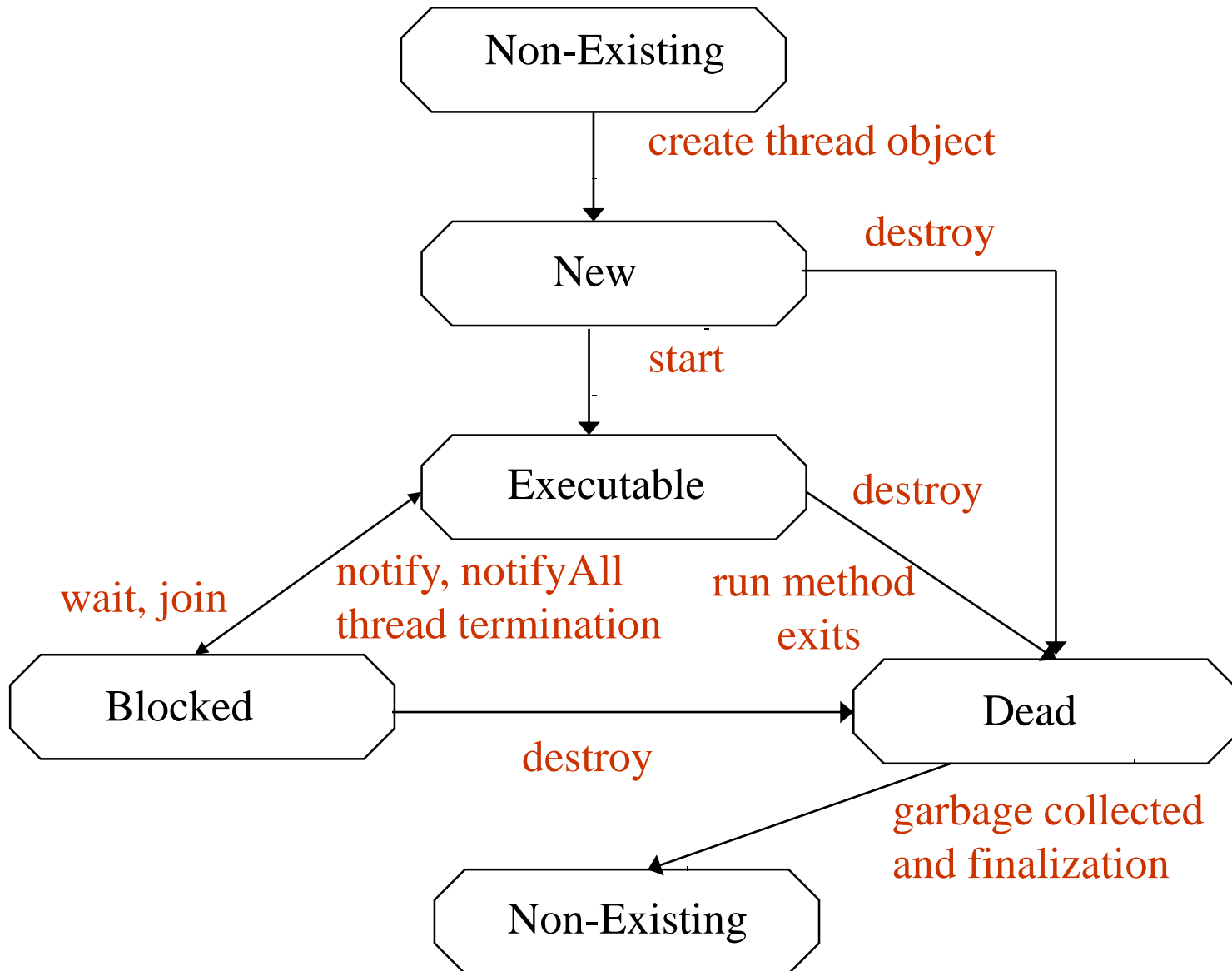
```
class ThreadEx1 {  
    public static void main(String [] args ) {  
    MyThread t = new MyThread();  
    Thread thr = new Thread(t);  
    thr.start();  
    System.out.println(" the main thread is now exiting ... ");  
    }  
}
```

Threads with Join

```
public class MyThread implements Runnable {  
    public void run() {  
        System.out.println(" this thread is now exiting ... ");  
    }}
```

```
class ThreadEx1 {  
    public static void main(String [] args ) {  
    MyThread t = new MyThread();  
    Thread thr = new Thread(t);  
    thr.start();  
    try {  
    thr.join();  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
    System.out.println(" the main thread is now exiting ... ");  
  
    }
```

Java Thread States



Thread States-II

- The thread is created when an object derived from the Thread class is created
- At this point, the thread is not executable — Java calls this the **new** state
- Once the `start` method has been called, the thread becomes eligible for execution by the scheduler
- If the thread calls the `wait` method in an Object, or calls the `join` method in another thread object, the thread becomes **blocked** and no longer eligible for execution
- It becomes executable as a result of an associated `notify` method being called by another thread, or if the thread with which it has requested a join, becomes **dead**

Thread States-III

- A thread enters the **dead** state, either as a result of the run method exiting (normally or as a result of an unhandled exception) or because its destroy method has been called