

1.3 Threads in Java

Java provides a *Thread* class for defining user threads. One way to define a thread is to define a class that extends (i.e. inherits from) class *Thread*.

```
class simpleThread extends Thread {
    public simpleThread(int ID) {myID = ID;}
    public void run() {System.out.println("Thread " + myID + " is running.");}
    private int myID;
}
public class javaConcurrentProgram {
    public static void main(String[] args) {
        simpleThread thread1 = new simpleThread(1);
        simpleThread thread2 = new simpleThread(2);
        thread1.start(); thread2.start(); // causes the run() methods to execute
    }
}
```

Listing 1.1 A simple concurrent Java program.

A second way to define a user thread in Java is to use the *Runnable* interface.

```
class simpleRunnable implements Runnable {
    public simpleRunnable(int ID) {myID = ID;}
    public void run() {System.out.println("Thread " + myID + " is running.");}
    private int myID;
}
public class javaConcurrentProgram2 {
    public static void main(String[] args) {
        Runnable r = new simpleRunnable(3);
        Thread thread3 = new Thread(r); // thread3 executed r's run() method
        thread3.start();
    }
}
```

Listing 1.2 Java's *Runnable* interface.

Class *simpleRunnable* could, if desired, extend some other class. This is important since a Java class cannot extend more than one other class. (A Java class can implement one or more interfaces, but extend only one class.)