

A **thread** is a program unit that is executed independently of the program. **Multithreading** refers to two or more threads executing concurrently.

- Threads are created and controlled by **Java.Lang.Thread** class.
- Threads *can* run in parallel, depending on the number of CPUs. If there is only one CPU, threads run systematically.
- Each thread runs for a short amount of time, called a time slice.

One way to create a thread in Java (including an example) is to:

- **Implement the Runnable interface (java.lang.Runnable)**

```
Public interface Runnable {
    void run();
}
class ThreadMan implements Runnable {
    Thread runner;
    public ThreadMan() { }
    public ThreadMan (String name) {
        // create a thread with a name and start it
        runner = new Thread(this,name);
        runner.start();
    }
    public void run() {
        // does something about the thread
        ...
    }
}
```

- The start() method of the Thread class starts a new thread that executes the run method of the associated Runnable object.
- The sleep(long millis) method puts the current thread to sleep for a certain time in milliseconds.
- When a thread is interrupted, an InterruptedException is generated. This needs to be caught in the run method and terminated.

```
try {
    // delay for 1 second
    Thread.currentThread().sleep(1000);
} catch (InterruptedException e) {
    //
}
```

- The thread terminates when *its* run method terminates.
- The run method should call the interrupted() method to check if its thread has terminated.