

CMPDS 109 3-2-11

Java Threads:

1st method: extend Thread class
in java.lang.

Ex. // SimpleThread.java

```
class SimpleThread extends Thread {
```

```
    private int id;
```

```
    private int delay;
```

```
    SimpleThread(int id, int delay) {
```

```
        this.id = id;
```

```
        this.delay = delay;
```

```
    }
```

```
public void run() {
```

2

```
    System.out.println("id + "started");  
    System.out.flush();
```

```
    for (int i=0; i<10; i++){
```

```
        try{
```

```
            sleep(delay);
```

```
        }
```

```
        catch (InterruptedException e) {
```

```
            System.out.println("interrupted: " + e);
```

```
        }
```

```
        System.out.println("id + " + i);
```

```
    }
```

```
    System.out.println("id + "finished")
```

```
}
```

```
}
```

```
// TwoThreads.java
class TwoThreads {
    public static void main (String[] args) {
        SimpleThread t1 = new SimpleThread(1, 1000);
        SimpleThread t2 = new SimpleThread(2, 1300);
        t1.start();
        t2.start();
    }
}
```

Can also implement the Runnable interface.

```
// SimpleThread2.java
```

```
class SimpleThread2 implements Runnable {
```

```
    private int id;
```

```
    private int delay;
```

```
    SimpleThread2(int id, int delay) {
```

```
        :
```

```
    }
```

```
    public void run() {
```

```
        System.out.println("id + " + "started");
```

```
        System.out.flush();
```

```
        for (int i = 0; i < 10; i++) {
```

```
            try {
```

```
                Thread.currentThread().sleep(delay);
```

```
            }
```

```
            catch (
```

```
                ) {
```

```
                :
```

```
            }
```

```
                System.out.println("id + " + " + i");
```

```
            }
        }
        System.out.println("id + " + "finished");
```

```
    }
```

```
// TwoThreads2.java
```

```
class TwoThreads2 {
```

```
    . . . . . main . . . . .
```

```
    Thread t1 = new Thread(new
```

```
        SimpleThread2(1, 1000));
```

```
    . . . . . t2 = new . . . . . (1, 1300));
```

```
    t1.start();
```

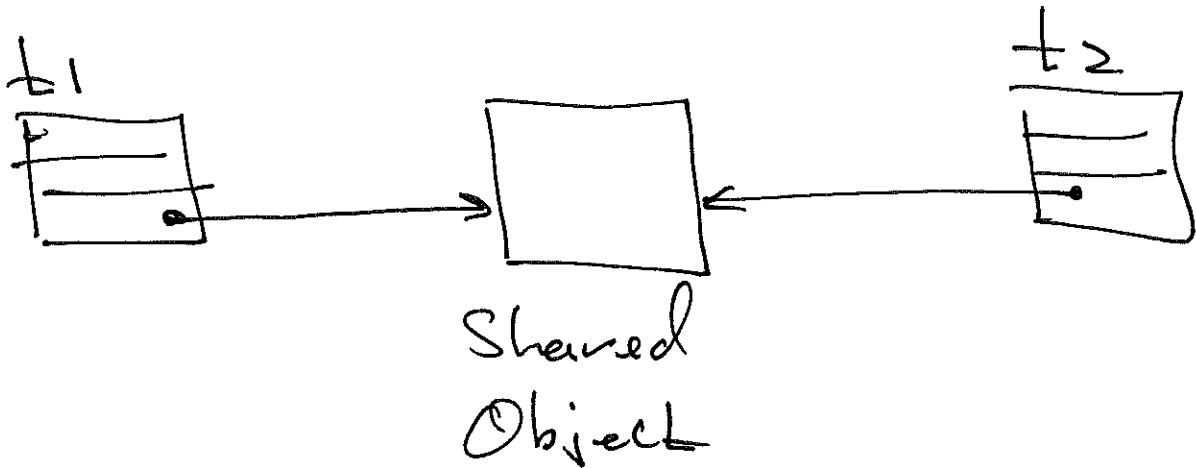
```
    t2.start();
```

```
}
```

How can threads communicate?

Three common ways:

- shared Objects
- messages (sockets... later)
- remote method invocation (RMI)



class Counter {

private int count;
private int limit;

Counter (int e, int m) {
 count = e;
 limit = m;
}

void click() {

 count = (count + 1) % limit;
}

int get() {

 return count;
}

}

```
// Racer.java
```

```
class Racer implements Runnable
```

```
    private int id;
```

```
    private Counter counter;
```

```
    Racer(int id, Counter counter) {
```

```
        this.id = id;
```

```
        this.counter = counter;
```

```
    }
```

```
    public void run() {
```

```
        System.out.println(id + ": start");
```

```
        for (int i = 0; i < 1000000; i++) {
```

```
            counter.click();
```

```
        }
```

```
        System.out.println(id + ": finish" +  
                            counter.get());
```

```
    }
```

```
}
```


// TwoRacers.java

class TwoRacers {

public . . . main(. . .) {

Counter c = new Counter(0, 10000000);

Thread t1 = new Thread(
new Racer(1, c));

Thread t2 = new Thread(
new Racer(2, c));

t1.start();

t2.start();

}

}