

# Philosopher Example

```
class PhilosophersShare {
    int [] state = new int[6];
    static final int THINKING = 0;
    static final int HUNGRY = 1;
    static final int EATING = 2;
    public Philosopher() {
        for(int i=0; i<6; i++) { state[i] = THINKING; }
    }
    public synchronized void pickUp(int id) {
        state[id] = HUNGRY;
        test(id);
        while(state[id] != EATING) {
            try { wait(); }
            catch(InterruptedException e) { }
        }
    }
    public synchronized void putDown(int id) {
        state[id] = THINKING;
        test((id + 5) % 6);
        test((id+1) % 6);
    }
    private test(int id) {
        if((state[(id + 5) % 6] != EATING) &&
            (state[id] == HUNGRY) &&
            (state[(id + 1) % 6] != EATING) {
            state[id] = EATING;
            notifyAll();
        }
    }
    public void thinkOrEat() {
        try { Thread.sleep((int)(2000 * Math.random())); }
        catch(InterruptedException e) { }
    }
}
```

# Philosopher Example

```
class Philosopher implements Runnable {
    private int id;
    private PhilosopherShare share;
    public Philosopher(int id, PhilosopherShare share) {
        this.id = id;
        this.share = share;
    }
    public void run() {
        for(;;) {
            System.out.println("Philosopher " + id + " is thinking");
            share.thinkOrEat();
            share.pickUp(id);
            System.out.println("Philosopher " + id + " is eating");
            share.thinkOrEat();
            share.putDown(id);
        }
    }
}
```

# Philosopher Example

```
class DiningPhilosophers {  
    public static void main(String [] args) {  
        PhilosopherShare share = new PhilosopherShare();  
        Philosopher [] p = new Philosopher[6];  
        for(int i=0; i<6; i++) {  
            p[i] = new Philosopher(i, share);  
        }  
        Thread [] t = new Thread[6];  
        for(int i=0; i<6; i++) {  
            t[i] = new Thread(p[i]);  
            t[i].start();  
        }  
    }  
}
```