```
package project4;

class Philosopher

  declare variable id;
  declare variable chopSticks;
  declare variable philosophers;

  contructor Philosopher()
  {

  }

function run()
  {
    while(true)
    {
      Thread.sleep (rand.nextInt(1000));
      chopSticks[id].acquire();       // pick left chopstick


      chopSticks[(id+1) % n].acquire();    // pick right chopstick

      Thread.sleep (rand.nextInt(1000));

      chopSticks[id].release();     // return left chopstick


      chopSticks[(id+1) % n].release(); // return right chopstick
      Thread.sleep (rand.nextInt(1000));
    }
  }



class project4

  declare integer PHILOSOPHERS_NO = 5;    // number of philosophers

  void main ()
  {
    create object chopSticks ;
    create object philosophers;
   }
```

```
class Semaphore

  declare integer value;

  contructor Semaphore()
  {
    this.value = value;
  }

  function acquire()
  {
    try
    {
      while (value <= 0)
        wait();
    }
    catch (InterruptedException e) {}

    --value;
  }

  function release()
  {
    ++value;
    notifyAll();
  }
}
```

2 possible solutions that you will implement in your simulation

Solution 1: Only allow 4 philosophers get chopsticks simultaneously.

Solution 2 : Add time sleep 1000 before `Philosopher pick left chopstick and after eat and think`