```
package SleepWait;



class GunFight {
   private int bullets = 40;



   synchronized public void fire(int bulletsToBeFired)
   {
      for (int i = 1; i <= bulletsToBeFired; i++) {
         if (bullets == 0) {
            System.out.println(i - 1
                  + " bullets fired and "
                  + bullets + " remains");
            System.out.println(
               "Invoking the wait() method");
            try {
               wait();
            }
            catch (InterruptedException e) {
               e.printStackTrace();
            }
            System.out.println(
               "Continuing the fire after reloading");
         }
```

```
            bullets--;
        }
        System.out.println(
            "The firing process is complete");
    }

    synchronized public void reload()
    {
        System.out.println(
            "Reloading the magazine and resuming "
            + "the thread using notify()");
        bullets += 40;
        notify();
    }
}

public class Wait1 extends Thread {
    public static void main(String[] args)
    {

        GunFight gf = new GunFight();


        new Thread() {
```

```
            @Override public void run() { gf.fire(60); }
        }.start();



        new Thread() {
            @Override public void run() { gf.reload(); }
        }.start();
    }
}
```