**Interthread Communication :--**

**Problem :--**

class First

{

int i;

synchronized void put(int num)

{

i=num;

System.out.println("Put: " + i);

}

synchronized int get()

{

System.out.println("Get: " + i);

return i;

}

}

class Putter implements Runnable

{

Thread t;

First f;

Putter(First z)

{

f=z;

t=new Thread(this);

t.start();

}

public void run()

{

int k=0;

while(f.i<=20)

f.put(++k);

}

}

class Getter implements Runnable

{

Thread t;

First f;

Getter(First z)

{

f=z;

t=new Thread(this);

t.start();

}

public void run()

{

while(f.i<=20)

f.get();

}

}

class Demo

{

public static void main(String args[])

{

First f1=new First();

new Putter(f1);

try

{

Thread.sleep(10);

}

catch(InterruptedException ie){}

new Getter(f1);

}

}

**Solution:--**

class First

{

int i;

boolean b=false;

synchronized void put(int num)

{

if(b)

{

try

{

wait();

}

catch(InterruptedException ie)

{

}

}

i=num;

System.out.println("Put: " + i);

notify();

b=true;

}

synchronized int get()

{

if(!b)

{

try

{

wait();

}

catch(InterruptedException ie)

{

}

}

System.out.println("Get: " + i);

notify();

b=false;

return i;

}

}

class Putter implements Runnable

{

Thread t;

First f;

Putter(First z)

{

f=z;

t=new Thread(this);

t.start();

}

public void run()

{

int k=0;

while(f.i<=20)

f.put(++k);

}

}

class Getter implements Runnable

{

Thread t;

First f;

Getter(First z)

{

f=z;

t=new Thread(this);

t.start();

}

public void run()

{

while(f.i<=20)

f.get();

}

}

class Demo

{

public static void main(String args[])

{

First f1=new First();

new Putter(f1);

try

{

Thread.sleep(2000);

}

catch(InterruptedException ie){}

new Getter(f1);

}

}