**PRACTICAL NO. 1**

**Aim: Process Communication**

public class Pc{

public static void main(String[] args){

Shop c = new Shop();

Producer p1 = new Producer(c, 1);

Consumer c1 = new Consumer(c, 1);

p1.start();

c1.start();

}

}

class Shop{

private int materials;

private boolean available = false;

public synchronized int get(){

while (available == false){

try {

wait();

}

catch (InterruptedException ie){}

}

available = false;

notifyAll();

return materials;

}

public synchronized void put(int value){

while (available == true){

try {

wait();

}

catch (InterruptedException ie){

ie.printStackTrace();;

}

}

materials = value;

available = true;

notifyAll();

}

}

class Consumer extends Thread{

private Shop Shop;

private int number;

public Consumer(Shop c, int number){

Shop = c;

this.number = number;

}

public void run(){

int value = 0;

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

value = Shop.get();

System.out.println("Consumer consumed "+this.number+" value and got: "+value);

}

}

}

class Producer extends Thread{

private Shop Shop;

private int number;

public Producer(Shop c, int number){

Shop = c;

this.number = number;

}

public void run(){

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

Shop.put(i);

System.out.println("Producer produced "+this.number+" value and put: "+i);

try{

sleep((int)(Math.random() \*100));

}

catch(InterruptedException ie){

ie.printStackTrace();

}

}

}

}

