1. class Clg{

Object clg, dept;

Clg(Object clg, Object dept)

{

this.clg=clg;

this.dept=dept;

}

}

class Resource\_1 extends Thread

{

Clg c;

Resource\_1(Clg c)

{

this.c=c;

}

public void run()

{

synchronized(c.clg){

System.out.println("New College");

try{

Thread.sleep(1000);

}

catch(Exception e){

}

synchronized(c.dept){

System.out.println("New Department");

}

}

}

}

class Resource\_2 extends Thread

{

Clg c;

Resource\_2(Clg c)

{

this.c=c;

}

public void run()

{

synchronized(c.clg){

System.out.println("College");

try{

Thread.sleep(1000);

}

catch (Exception e){}

}

synchronized(c.dept){

System.out.println("Department");

}

}

}

class DeadLock{

public static void main(String... args)

{

Object clg=new Object();

Object dept=new Object();

Clg c=new Clg(clg, dept);

Resource\_1 r1=new Resource\_1(c);

Resource\_2 r2=new Resource\_2(c);

Thread t1=new Thread(r1);

Thread t2=new Thread(r2);

t1.start();

t2.start();

}

}

2. class Manager extends Thread{

StringBuffer s;

Manager(){

s = new StringBuffer();

}

public void run(){

synchronized(s){

for(int i=0;i<=20;i++){

s.append(i+", ");

}

s.notify();

System.out.println("Manager notifying Employee to start working");

}

}

}

class Employee extends Thread{

Manager mgr;

Employee(Manager mgr){

this.mgr = mgr;

}

public void run(){

synchronized(mgr.s){

try{

System.out.println("Employee is waiting for Manager to notify the Message");

mgr.s.wait();

}catch(InterruptedException ie){

}

System.out.println(mgr.s);

}

}

}

class ThreadEx {

public static void main(String... args){

Manager mgr = new Manager();

Employee emp = new Employee(mgr);

Thread t1 = new Thread(mgr);

Thread t2 = new Thread(emp);

t2.start();

t1.start();

}

}

3. a) import java.io.\*;

import java.net.\*;

class Server{

public static void main(String args[]) throws Exception

{

ServerSocket ss = new ServerSocket(888);

Socket s = ss.accept();

System.out.println("Connection established");

PrintStream ps = new PrintStream(s.getOutputStream());

BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));

BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));

while (true) {

String str, str1;

while ((str = br.readLine()) != null)

{

System.out.println(str);

str1 = kb.readLine();

ps.println(str1);

}

ps.close();

br.close();

kb.close();

ss.close();

s.close();

}

}

3

class Clg{

Object clg, dept;

Clg(Object clg, Object dept)

{

this.clg=clg;

this.dept=dept;

}

}

class Resource\_1 extends Thread

{

Clg c;

Resource\_1(Clg c)

{

this.c=c;

}

public void run()

{

synchronized(c.clg){

System.out.println("New College");

try{

Thread.sleep(1000);

}

catch(Exception e){

}

synchronized(c.dept){

System.out.println("New Department");

}

}

}

}

class Resource\_2 extends Thread

{

Clg c;

Resource\_2(Clg c)

{

this.c=c;

}

public void run()

{

synchronized(c.clg){

System.out.println("College");

try{

Thread.sleep(1000);

}

catch (Exception e){}

}

synchronized(c.dept){

System.out.println("Department");

}

}

}

class DeadLock{

public static void main(String... args)

{

Object clg=new Object();

Object dept=new Object();

Clg c=new Clg(clg, dept);

Resource\_1 r1=new Resource\_1(c);

Resource\_2 r2=new Resource\_2(c);

Thread t1=new Thread(r1);

Thread t2=new Thread(r2);

t1.start();

t2.start();

}

}class Clg{

Object clg, dept;

Clg(Object clg, Object dept)

{

this.clg=clg;

this.dept=dept;

}

}

class Resource\_1 extends Thread

{

Clg c;

Resource\_1(Clg c)

{

this.c=c;

}

public void run()

{

synchronized(c.clg){

System.out.println("New College");

try{

Thread.sleep(1000);

}

catch(Exception e){

}

synchronized(c.dept){

System.out.println("New Department");

}

}

}

}

class Resource\_2 extends Thread

{

Clg c;

Resource\_2(Clg c)

{

this.c=c;

}

public void run()

{

synchronized(c.clg){

System.out.println("College");

try{

Thread.sleep(1000);

}

catch (Exception e){}

}

synchronized(c.dept){

System.out.println("Department");

}

}

}

class DeadLock{

public static void main(String... args)

{

Object clg=new Object();

Object dept=new Object();

Clg c=new Clg(clg, dept);

Resource\_1 r1=new Resource\_1(c);

Resource\_2 r2=new Resource\_2(c);

Thread t1=new Thread(r1);

Thread t2=new Thread(r2);

t1.start();

t2.start();

}

}