1. Create a class Table including a method printable which prints multiplication table of a given value . Create two threads which prints multiplication table of 5 and 100 by calling the same function on same object. Test the threads

class Table{

void tables(int a) {

try {

Thread.*sleep*(2000);

}catch(InterruptedException e){

System.*out*.println("Thread Interrupted");

}

for(int i=1;i<=10;i++) {

System.*out*.println(a + "X" + i + "=" + (i\*a));

}

}

}

class Multiplytable implements Runnable{

Thread t;

Table tb;

int m;

Multiplytable(Table tble,int a){

tb=tble;

m=a;

t= new Thread(this);

t.start();

}

public void run() {

synchronized(tb) {

tb.tables(m);

}

}

}

public class Synchronization {

public static void main(String args[]) {

Table c= new Table();

Multiplytable ob1= new Multiplytable(c,5);

Multiplytable ob2= new Multiplytable(c,100);

try {

ob1.t.join();

ob2.t.join();

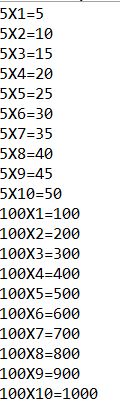
}catch(InterruptedException e){

System.*out*.println("Main thread Interrupted.");

}

}

}



1. Create two class Car\_Owner and Car\_Mechanic . Make them threads. The two threads tries to access an object of Car\_queue class. Car\_queue involves get and put function based on the availability of mechanic. Car\_owner should put the request for car service and Car mechanic should get the request posed by car owner one at a time.

class Car\_Queue{

boolean value=false;

synchronized void get() {

while(!value) {

try {

wait();

}catch(InterruptedException e) {

System.*out*.println(" Thread Interrupted");

}

}

value=false;

System.*out*.println("request RECEIVED.");

notify();

}

synchronized void put() {

while(value) {

try {

wait();

}catch(InterruptedException e) {

System.*out*.println(" Thread Interrupted");

}

}

value=true;

System.*out*.println("request PUT.");

notify();

}

}

class Car\_owner implements Runnable{

Car\_Queue q;

Thread t;

Car\_owner(Car\_Queue cq){

q=cq;

t=new Thread(this);

t.start();

}

public void run() {

q.put();

}

}

class Car\_mech implements Runnable{

Car\_Queue q;

Thread t;

Car\_mech(Car\_Queue cq){

q=cq;

t=new Thread(this);

t.start();

}

public void run() {

q.get();

}

}

public class InterthreadCom {

public static void main(String args[]) {

Car\_Queue cq =new Car\_Queue();

Car\_owner co= new Car\_owner(cq);

Car\_mech cm= new Car\_mech(cq);

try {

co.t.join();

cm.t.join();

}catch(InterruptedException e) {

System.*out*.println(" Main Thread Interrupted");

}

}

}

