**Term-work 3**

**Complete problem definition of the Term-work**

**Write** a Java program to simulate LOST UPDATE or INCONSISTENT READ Transaction issues of database using MULTITHREADING features of JAVA.

Also write a java program to control the above concurrency issue. Output of the program to be displayed on the screen as well as to be written in a file of user choice.

**Expected Learning**: Multithreading and Streams of java language

**Source code/Program**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author kishan

\*/

import java.io.\*;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;

import java.util.concurrent.TimeUnit;

class Counter1 {

int count = 0;

public void increment() {

count = count + 1;

}

public int getCount() {

return count;

}

}

class SynchronizedCounter1 {

private int count = 0;

public synchronized void increment() {

count = count + 1;

}

public int getCount() {

return count;

}

}

public class LostUpdate2 {

public static void main(String args []) throws InterruptedException, IOException {

int unsyncCount;

int syncCount;

String c;

char c3[] = new char [10];

ExecutorService exunsync = Executors.newFixedThreadPool(10);

Counter1 counter1 = new Counter1();

for(int i = 0;i < 1000; i++){

exunsync.submit(() -> counter1.increment());

}

exunsync.shutdown();

exunsync.awaitTermination(60, TimeUnit.SECONDS);

unsyncCount = counter1.getCount();

System.out.println("Total Count at end UNSYNC: " +unsyncCount);

ExecutorService exsync = Executors.newFixedThreadPool(10);

SynchronizedCounter1 counter2 = new SynchronizedCounter1();

for(int i = 0;i < 1000; i++){

exsync.submit(() -> counter2.increment());

}

exsync.shutdown();

exsync.awaitTermination(60, TimeUnit.SECONDS);

syncCount = counter2.getCount();

System.out.println("Total Count at end SYNC: " +syncCount);

Writer wr = new FileWriter("output.txt");

FileReader fin1 = new FileReader("output.txt");

try{

wr.append(String.valueOf(unsyncCount));

wr.append('\n');

wr.append(String.valueOf(syncCount));

}catch(IOException e){

System.out.println("Exception "+e);

}

wr.close();

System.out.println("Values of count after unsync and sync");

try{

int c1;

while((c1=fin1.read())!=-1)

System.out.print((char)c1);

System.out.println();

}catch(IOException e){

System.out.println("Exception "+e);

}

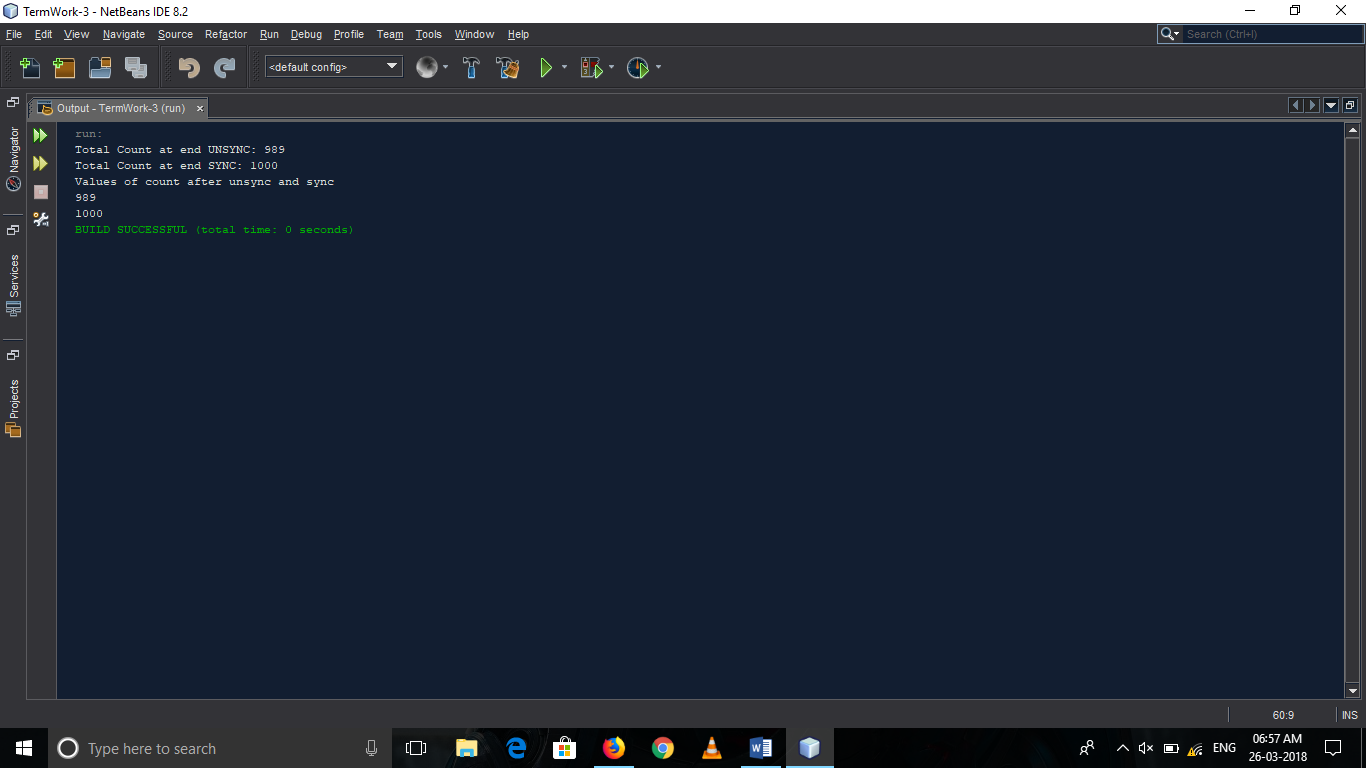
fin1.close();

}

}

**Output with Screenshot**

//Output on console



//Output from file

