

## Practical-4

**Q.Write a program to solve Classical Problems of Synchronization using Mutex and Semaphore**

```
import java.util.*;
import java.util.concurrent.Semaphore; public
class Main {
static int mutex=1; static int
database=1; static int Read_Count=0;
static void Reader() throws Exception
{ while(true)
{
mutex=wait(mutex);
Read_Count=Read_Count+1;
if(Read_Count==1){ database=signal(database);
}
mutex=signal(mutex);
System.out.println(Read_Count+ " User Reading the Data.....");
mutex=wait(mutex); Read_Count=Read_Count-1;
if(Read_Count==0)
{
database=signal(database);
}
mutex=signal(mutex);
System.out.println("Reading Finished!!!!!!");
break; } }
static int wait(int mutex)
{
while(mutex<=0)
break ;
mutex=mutex-1;
return mutex; }
static int signal(int database)
{ database=database+1;
return database;
}
static void Writer() throws Exception
{ while(true)
{
database=wait(database);
System.out.println("Writing on the database....."); database=signal(database);
System.out.println("Writing Finished!!!!!!");
break; } }
public static void main(String[] args)throws Exception {
Writer();
```

```
Reader();  
Reader();  
}  
}
```

#Output

Writing on the database.....

Writing Finished!!!!.

1 User Reading the Data..... Reading  
Finished!!!!.

1 User Reading the Data..... Reading  
Finished!!!!.