

## LAB # 5

### Inter-Thread Communication:

#### OBJECTIVE

Develop an inter-thread user communication program by using synchronization.

#### Lab Task:

1. Design a simple program of concurrency by implementing the scenario of two account holders in a joint bank account. (Hint: Total amount will be 50000, if ‘user A’ wants to withdraw 45,000 and ‘user B’ wants to withdraw 20,000) Apply mechanism of synchronization e.g. Block or Method for handling accessibility of multi-threads:

```
Main.java

1 - class JointAccount {
2     private int balance = 50000;
3     public synchronized void withdraw(String user, int amount) {
4
5         System.out.println(user + " is trying to withdraw: " + amount);
6
7         if (balance >= amount) {
8             System.out.println("Sufficient balance. Processing...");
9             try { Thread.sleep(1000); }
10            catch (InterruptedException e) {}
11
12             balance -= amount;
13             System.out.println(user + " successfully withdrew " + amount);
14         } else {
15             System.out.println("INSUFFICIENT BALANCE for " + user);
16         }
17         System.out.println("-----");
18     }
19 - class UserThread extends Thread {
20     JointAccount account;
21     String userName;
22     int amount;
23 - public UserThread(JointAccount acc, String name, int amt) {
24     this.account = acc;
25     this.userName = name;
26     this.amount = amt;
27 }
28 - public void run() {
29     account.withdraw(userName, amount);
30 }
31 - public class Main {
32     public static void main(String[] args) {
33         JointAccount account = new JointAccount();
34         UserThread userA = new UserThread(account, "Abdul Ahad", 45000);
35         UserThread userB = new UserThread(account, "Talha", 20000);
36         userA.start();
37         userB.start();
38     }
39 }
```

#### OUTPUT:

```
Abdul Ahad is trying to withdraw: 45000
Sufficient balance. Processing...
Abdul Ahad successfully withdrew 45000
Remaining Balance: 5000
-----
Talha is trying to withdraw: 20000
INSUFFICIENT BALANCE for Talha
Remaining Balance: 5000
-----
```

2. Create an inter thread communication program of printer job by implementing two threads, one for calculating the remaining pages in printer tray and other one will print the pages that are pending on queue. (Hint: If total pages are 10 and user sends job for 15 pages than print thread will be on wait and will be notified once available pages are equal or greater than printing pages).

```

Main.java
1  class Printer {
2      private int trayPages = 10;
3      public synchronized void printPages(int pagesToPrint) {
4          System.out.println("Print Job: Requested " + pagesToPrint + " pages.");
5          while (trayPages < pagesToPrint) {
6              System.out.println("Not enough pages! Available: " + trayPages +
7                               " | Required: " + pagesToPrint);
8              System.out.println("Print thread is WAITING...\n");
9              try {
10                  wait(); // wait until tray is refilled
11              } catch (InterruptedException e) {
12                  e.printStackTrace();
13              }
14          }
15          System.out.println("Printing started...");
16          trayPages -= pagesToPrint;
17          System.out.println("Printing completed!");
18          System.out.println("Remaining pages in tray: " + trayPages);
19      }
20      public synchronized void refillTray(int pages) {
21          System.out.println("Refill Thread: Adding " + pages + " pages to tray...");
22          trayPages += pages;
23          System.out.println("Tray now contains: " + trayPages + " pages\n");
24          notify();
25      }
26      class PrintJobThread extends Thread {
27          Printer printer;
28          public PrintJobThread(Printer p) {
29              this.printer = p;
30          }
31          public void run() {
32              printer.printPages(15);
33          }
34      }
35      class TrayRefillThread extends Thread {
36          Printer printer;
37          public TrayRefillThread(Printer p) {
38              this.printer = p;
39          }
40          public void run() {
41              try {
42                  Thread.sleep(2000);
43              } catch (InterruptedException e) {}
44              printer.refillTray(10);
45          }
46  }
47  public class Main {
48      public static void main(String[] args) {
49          Printer printer = new Printer();
50          PrintJobThread printJob = new PrintJobThread(printer);
51          TrayRefillThread refill = new TrayRefillThread(printer);
52          printJob.start();
53          refill.start();
54      }
55  }

```

## OUTPUT:

```

Print Job: Requested 15 pages.
Not enough pages! Available: 10 | Required: 15
Print thread is WAITING...

Refill Thread: Adding 10 pages to tray...
Tray now contains: 20 pages

Printing started...
Printing completed!
Remaining pages in tray: 5

```