

## CNT 4714 – Project – Spring 2023

**Title:** “Project 2: An Application Employing Synchronized/Cooperating Multiple Threads In Java Using Locks – A Banking Simulator”

**Points:** 100 points

**Due Date:** Sunday February 12, 2023 by 11:59 pm (WebCourses time)

**Objectives:** To develop an application which requires cooperating, synchronized multiple threads of execution.

**Description:** In this programming assignment you will simulate the deposits and withdrawals made to a fictitious bank account (I'll let you use my real bank account if you promise to make only deposits! ☺). In this case the deposits and withdrawals will be made by user agents (synchronized threads). Synchronization is required for two reasons – (1) mutual exclusion (updates cannot be lost) and (2) because a withdrawal cannot occur if the amount of the withdrawal request is greater than the current balance in the account. This means that access to the account (the shared object) must be synchronized. This application requires cooperation and communication amongst the various agents (cooperating synchronized threads). (In other words, this problem is similar to the producer/consumer problem where there is more than one producer and more than one consumer process active simultaneously.) If a withdrawal agent attempts to withdraw an amount greater than the current balance in the account – then it must block itself and wait until a depositing agent has added money to the account before it can try again. As we covered in the lecture notes, this will require that the depositing agents signal all waiting withdrawing agents whenever a deposit is completed.

1. You should have **five depositor** agents (threads) and **ten withdrawal** agents (threads), and **one auditor** agent (thread) simultaneously executing. Use a `FixedThreadPool()` and an `Executor` object to control the threads.
2. To keep things relatively simple, as well as to see immediate results from a series of transactions (deposits and withdrawals), assume that deposits are made in amounts ranging from \$1 to \$500 (whole dollars only) and withdrawals are made in amounts ranging from \$1 to \$99 (again, whole dollars only). Since we have more withdrawal threads than depositor threads, the account balance should constantly decrease over time. This will lead to withdrawal agents repeatedly blocking for insufficient funds. Start the simulation with a balance of \$0.

3. Once a depositor agent (thread) has deposited into the account, put it to sleep for few milliseconds (randomly generate this number – don't use a constant sleep time) or so (depends a little bit on the speed of your system as to how long you will want to sleep the depositing threads - basically we want to ensure a lot more withdrawals than deposits) to allow other agents to execute. This is the only situation in which a deposit agent will block.
4. For withdrawal agents, things will be a bit different depending on whether you are working on a single or multi-core processor.
  - a. For single core processors, once a withdrawal agent has withdrawn funds from the account, have it yield the processor unit. Since the agent is giving up the processor voluntarily, it will be unlikely to run again (attempt a second withdrawal in a row), before another agent runs. Note however, that it does not prevent it from running again, if all other withdrawal agents are blocked and all depositors are sleeping, it will run again. So occasional back-to-back runs of withdrawal agents might occur (see below).
  - b. For multi-core processors, once a withdrawal agent has executed, have it sleep for some random period of time (again, a few milliseconds should be fine). Depending on which core a thread is executing, yielding the CPU won't ensure that the same thread will not run again immediately. While, sleeping the thread will also not ensure that it will not run two or more times in succession, it is less likely to do so in the multi-core environment.
  - c. What we don't want to happen is a single withdrawal agent gaining the CPU and then executing a long sequence of withdrawal operations. Recall though, that withdrawal agents block if they attempt to withdraw more than the current balance in the account.
  - d. Similarly, we don't want depositor agents monopolizing the CPU either and causing the balance in the account to grow continuously. This would most likely occur when the withdrawal agents are sleeping too long in comparison to the average sleep time of the depositing agents. See page 9 for an illustration of this.
5. Assume all depositor and withdrawal agents have the same priority. **Do not** give different priority to depositor and withdrawal agents (threads). The auditor agent will also have normal priority and will simply run less frequently than the depositor and withdrawal threads (i.e., the auditor thread will sleep longer between runs than either depositors or withdrawals). See below.
6. The output from your program must look reasonably similar to the sample output shown below. The simulation output should show the action of each agent along

with the account balance produced by the agent's transaction and the transaction number.

7. **Do not put the threads into a counted loop for your simulation.** In other words, the `run()` method for all threads should be an infinite loop. Just stop the simulation from your IDE after a few seconds.
8. **Do not use the Java synchronized statement.** I want you to handle the locking and signaling yourself. No monitors!
9. You must utilize a reentrant lock from the `java.util.concurrent.locks` package for implementing your locking protocols. We will specify no fairness policy for this application. **Do not create your own lock using a Boolean or any other type of variable.**
10. The Money Laundering Suppression Act, enacted by Congress in 1994, is a policy regulation that requires banking institutions to file currency transaction reports (CTRs) with the federal government (Department of the Treasury) for any deposits of \$10,000 or more into a bank account. You are going to simulate this process by flagging any depositing transaction with a deposit value greater than \$350.00 and any withdrawal amount greater than \$75.00. You will flag the transaction in the normal output of the simulation as well as making an entry into a transaction log file which will keep track of all flagged transactions independently of the simulation. Each entry in the flagged transaction file will contain the transaction details, a timestamp, and the transaction number. See below for more details.
11. Every transaction made by a depositor or withdrawal agent will have a transaction number (an integer initialized to 1). This transaction number is printed out in the simulation run with each completed transaction. See output examples below.
12. The auditor agent (there is only 1 of these), simply verifies the current balance in the account at random intervals. The auditor agent does not make transactions on the account and does not affect the transaction number sequence. The auditor agent simply print the current account balance into the simulation run and keeps track of the number of transactions that have executed since the last auditor execution. Note that the auditor should run much less frequently than either depositor or withdrawal agents.

**References:**

Notes: Lecture Notes for Multithreaded Applications.

**Restrictions:**

Your source files shall begin with comments containing the following information:

*/\* Name:*

**Course: CNT 4714 Spring 2023**

**Assignment title: Project 2 – Synchronized, Cooperating Threads Under Locking**

**Due Date: February 12, 2023**

*\*/*

**Input Specification:** Internal to the program.

**Output Specification:** Console based. Your output should appear reasonably similar to the output shown below.

**Deliverables:**

- (1) Zip up all of your .java files and submit them via WebCourses no later than 11:59pm Sunday February 12, 2023.
- (2) Include a sufficient number of screen shots that illustrate the execution of your synchronized threaded application (the simulation output). See below for some representative examples. You can either do a screen shot of the console window like I did below or redirect your output to a file and take a screen shot from an editor. Your screenshot should illustrate all the different types of output we expect to see. Include as many screen shots as necessary. Label all screenshots clearly. Most IDEs will allow you to redirect the console output to a file. A copy of this redirected output file would be the preferred technique for illustrating the execution of your simulation. As far as I am aware, NetBeans does not allow for console redirection, but most other IDEs do.
- (3) Include a copy of your transaction log file that matches the simulation output (see below for explanation).

**Additional Information:**

Shown below are some examples of the output from this program to help illustrate how your application is to operate and display the results. The last page illustrates execution runs that you do not want to produce.

```

eclipse-workspace-2022-12 - CNT 4714 - Project 2 - Spring 2023/src/project2V2/ABankAccount.java - Eclipse IDE

@ Javadoc Declaration Console X
<terminated> ABankingSimulator [Java Application] /Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java (Jan 29, 2023, 7:54:37 PM - 7:54:40 PM) [pid: 65936]

Deposit Agents Withdrawal Agents Balance Transaction Number
-----|-----|-----|-----|-----|
Agent WT3 withdraws $53      (*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!
Agent WT4 withdraws $58      (*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!
Agent WT5 withdraws $45      (*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!
Agent WT6 withdraws $90      (*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!
Agent DT3 deposits $497      (+) Balance is $497          1
*** Flagged Transaction - Depositor Agent DT3 Made A Deposit In Excess Of $350.00 USD - See Flagged Transaction Log.

Agent WT2 withdraws $94      (-) Balance is $403          2
*** Flagged Transaction - Withdrawal Agent WT2 Made A Withdrawal In Excess Of $75.00 USD - See Flagged Transaction Log.

Agent DT4 deposits $64      (+) Balance is $467          3
Agent WT1 withdraws $60      (-) Balance is $407          4
Agent WT7 withdraws $84      (-) Balance is $323          5
*** Flagged Transaction - Withdrawal Agent WT7 Made A Withdrawal In Excess Of $75.00 USD - See Flagged Transaction Log.

Agent DT1 deposits $12      (+) Balance is $331          6
Agent WT9 withdraws $9       (-) Balance is $322          7
Agent DT0 deposits $213      (+) Balance is $535          8
Agent DT2 deposits $380      (+) Balance is $915          9
Agent WT0 withdraws $97      (-) Balance is $818          10
*** Flagged Transaction - Depositor Agent DT2 Made A Deposit In Excess Of $350.00 USD - See Flagged Transaction Log.

Agent WT0 withdraws $97      (-) Balance is $818          11
*** Flagged Transaction - Withdrawal Agent WT0 Made A Withdrawal In Excess Of $75.00 USD - See Flagged Transaction Log.

*****
AUDITOR FINDS CURRENT ACCOUNT BALANCE TO BE: $818           Number of transactions since last audit is: 11
*****

Agent WT9 withdraws $69      (-) Balance is $749          12
Agent WT3 withdraws $74      (-) Balance is $675          13
Agent WT3 withdraws $20      (-) Balance is $655          14
Agent WTB withdraws $39      (-) Balance is $616          15
Agent WT6 withdraws $8       (-) Balance is $608          16

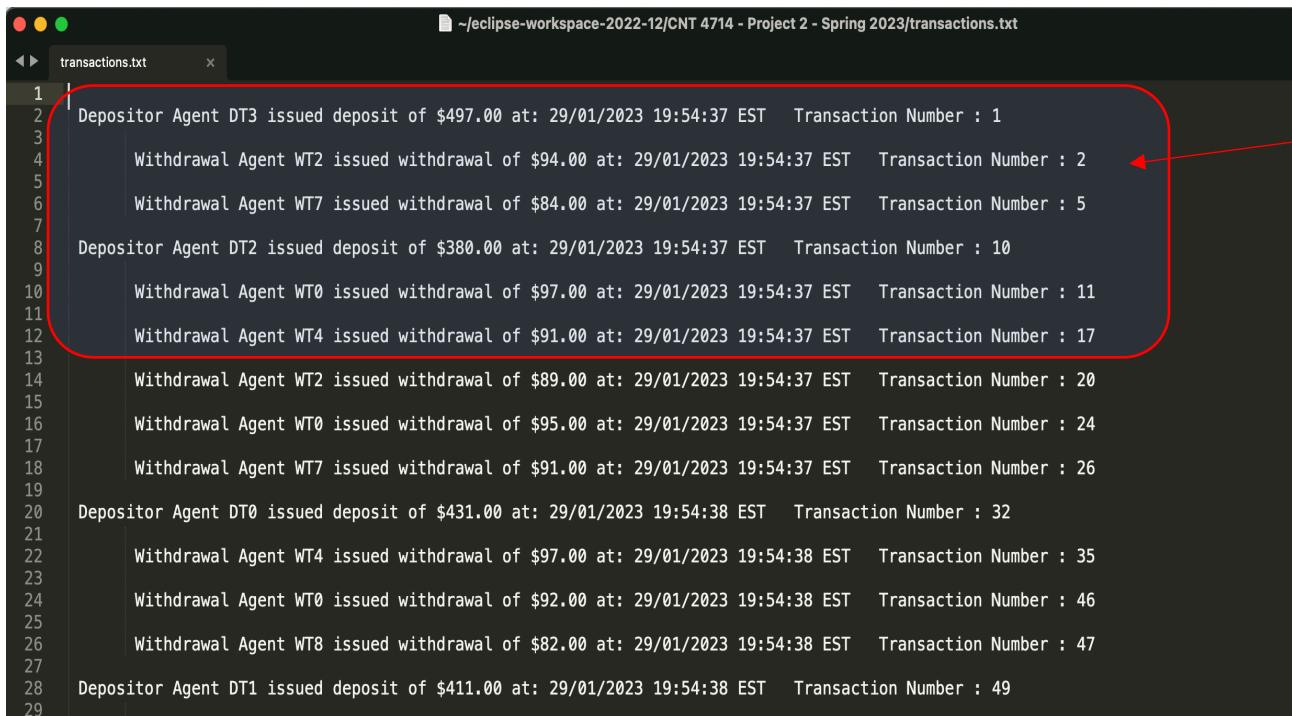
*****
AUDITOR FINDS CURRENT ACCOUNT BALANCE TO BE: $608           Number of transactions since last audit is: 5
*****

Agent WT4 withdraws $91      (-) Balance is $517          17
*** Flagged Transaction - Withdrawal Agent WT4 Made A Withdrawal In Excess Of $75.00 USD - See Flagged Transaction Log.

Agent WT9 withdraws $10      (-) Balance is $507          18
Agent WT6 withdraws $14      (-) Balance is $493          19
Agent WT2 withdraws $89      (-) Balance is $404          20

```

The transaction log file showing the first six flagged transactions in the execution run shown above.



```
~ /eclipse-workspace-2022-12/CNT 4714 - Project 2 - Spring 2023/transactions.txt
transactions.txt

1 Depositor Agent DT3 issued deposit of $497.00 at: 29/01/2023 19:54:37 EST Transaction Number : 1
2 Withdrawal Agent WT2 issued withdrawal of $94.00 at: 29/01/2023 19:54:37 EST Transaction Number : 2
3 Withdrawal Agent WT7 issued withdrawal of $84.00 at: 29/01/2023 19:54:37 EST Transaction Number : 5
4 Depositor Agent DT2 issued deposit of $380.00 at: 29/01/2023 19:54:37 EST Transaction Number : 10
5 Withdrawal Agent WT0 issued withdrawal of $97.00 at: 29/01/2023 19:54:37 EST Transaction Number : 11
6 Withdrawal Agent WT4 issued withdrawal of $91.00 at: 29/01/2023 19:54:37 EST Transaction Number : 17
7 Withdrawal Agent WT2 issued withdrawal of $89.00 at: 29/01/2023 19:54:37 EST Transaction Number : 20
8 Withdrawal Agent WT0 issued withdrawal of $95.00 at: 29/01/2023 19:54:37 EST Transaction Number : 24
9 Withdrawal Agent WT7 issued withdrawal of $91.00 at: 29/01/2023 19:54:37 EST Transaction Number : 26
10 Depositor Agent DT0 issued deposit of $431.00 at: 29/01/2023 19:54:38 EST Transaction Number : 32
11 Withdrawal Agent WT4 issued withdrawal of $97.00 at: 29/01/2023 19:54:38 EST Transaction Number : 35
12 Withdrawal Agent WT0 issued withdrawal of $92.00 at: 29/01/2023 19:54:38 EST Transaction Number : 46
13 Withdrawal Agent WT8 issued withdrawal of $82.00 at: 29/01/2023 19:54:38 EST Transaction Number : 47
14 Depositor Agent DT1 issued deposit of $411.00 at: 29/01/2023 19:54:38 EST Transaction Number : 49
```

\*\*\* Flagged Transaction - Withdrawal Agent WT0 Made A Withdrawal In Excess Of \$75.00 USD - See Flagged Transaction Log.

Agent WT3 withdraws \$49      (-) Balance is \$204  
Agent WT7 withdraws \$91      (-) Balance is \$113

All withdrawal threads are blocked. No withdrawal thread can run until a depositor thread runs.

\*\*\* Flagged Transaction - Withdrawal Agent WT7 Made A Withdrawal In Excess Of \$75.00 USD - See Flagged Transaction Log.

Agent WT8 withdraws \$44	(-) Balance is \$69	27
Agent WT3 withdraws \$40	(-) Balance is \$29	28
Agent WT4 withdraws \$52	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT9 withdraws \$3	(-) Balance is \$26	29
Agent WT2 withdraws \$17	(-) Balance is \$9	30
Agent WT9 withdraws \$4	(-) Balance is \$5	31
Agent WT6 withdraws \$65	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT5 withdraws \$48	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT8 withdraws \$19	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT7 withdraws \$85	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT3 withdraws \$90	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT2 withdraws \$22	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT0 withdraws \$38	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT9 withdraws \$48	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT1 withdraws \$33	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	

\*\*\*\*\*

AUDITOR FINDS CURRENT ACCOUNT BALANCE TO BE: \$5

Number of transactions since last audit is: 15

\*\*\*\*\*

Agent DT0 deposits \$431      (+) Balance is \$436      32

\*\*\* Flagged Transaction - Depositor Agent DT0 Made A Deposit In Excess Of \$350.00 USD - See Flagged Transaction Log.

Agent WT7 withdraws \$51	(-) Balance is \$385	33
Agent WT6 withdraws \$46	(-) Balance is \$339	34
Agent WT4 withdraws \$97	(-) Balance is \$242	35

Agent WT8 withdraws \$82      (-) Balance is \$29      47

\*\*\* Flagged Transaction - Withdrawal Agent WT8 Made A Withdrawal In Excess Of \$75.00 USD - See Flagged Transaction Log.

Agent WT2 withdraws \$23	(-) Balance is \$6	
Agent WT4 withdraws \$32	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT0 withdraws \$18	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT5 withdraws \$79	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT3 withdraws \$13	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT8 withdraws \$12	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT9 withdraws \$63	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT6 withdraws \$82	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT7 withdraws \$85	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT1 withdraws \$87	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	
Agent WT2 withdraws \$66	(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!	

Withdrawal thread WT2 runs two times in a row. This is ok. It may happened from time to time.

Agent DT1 deposits \$411      (+) Balance is \$417      49

\*\*\* Flagged Transaction - Depositor Agent DT1 Made A Deposit In Excess Of \$350.00 USD - See Flagged Transaction Log.

Agent WT4 withdraws \$1	(-) Balance is \$416	50
Agent WT2 withdraws \$39	(-) Balance is \$377	51
Agent WT2 withdraws \$59	(-) Balance is \$318	52
Agent WT3 withdraws \$1	(-) Balance is \$317	53

\*\*\*\*\*

\* \* \* Flagged Transaction - Withdrawal Agent WTO Made A Withdrawal

Agent WT4 withdraws \$66  
Agent WT1 withdraws \$70  
Agent WT3 withdraws \$20  
Agent WT9 withdraws \$20  
Agent WT1 withdraws \$47  
Agent WT4 withdraws \$69  
Agent WT5 withdraws \$64  
Agent WT8 withdraws \$35  
Agent WT7 withdraws \$69  
Agent WT0 withdraws \$35  
Agent WT6 withdraws \$72  
Agent WT2 withdraws \$71

Agent DT2 deposits \$129

Agent WT6 withdraws \$3  
Agent WT5 withdraws \$68  
Agent WT7 withdraws \$94  
Agent WT3 withdraws \$96  
Agent WT2 withdraws \$56  
Agent WT9 withdraws \$26  
Agent WT8 withdraws \$49  
Agent WT8 withdraws \$42  
Agent WT2 withdraws \$20  
Agent WT4 withdraws \$34  
Agent WT6 withdraws \$39  
Agent WT5 withdraws \$3  
Agent WT1 withdraws \$43

Agent DT2 deposits \$125

Agent WT1 withdraws \$5  
Agent WT2 withdraws \$69  
Agent WT0 withdraws \$40

1. Withdrawal agent WT6 is blocked (along with other withdrawal agents).

(-) Balance is \$6  
\*\*\*\*\* WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!  
\*\*\*\*\* WITHD  
\*\*\*\*\* WITHD  
\*\*\*\*\* WITHD  
\*\*\*\*\* WITHD  
\*\*\*\*\* WTTD

2. Deposit agent DT2 makes deposit and signals all waiting withdrawal agents.

```
(+) Balance is $135
(-) Balance is $132
(-) Balance is $64
(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!
(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!
(-) Balance is $8
(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!
(*****) WITHDRAWAL BLOCKED - INSUFFICIENT FUNDS!!!
```

3. Withdrawal agents WT6 and WT5 manage to make new withdrawals, but bleed the account balance down so that subsequent withdrawal agents once again find an insufficient balance and become blocked again.

(-) Balance is \$16

We don't want to see this sort of scenario where the depositors are monopolizing the account. Indication is the depositor threads aren't sleeping long enough or the withdrawal threads are sleeping too long.

**Balance just continues to grow and no blocking ever occurs.**

```

eclipse-workspace-2022-12 - CNT 4714 - Project 2 - Spring 2023/src/project2V2/Depositor.java - Eclipse IDE
@ Javadoc Declaration Console X
<terminated> ABankingSimulator [Java Application] /Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java (Jan 29, 2023, 8:22:31 PM – 8:22:32 PM) [pid: 66062]

Agent DT3 deposits $136          (+) Balance is $6361      39
Agent DT2 deposits $263          (+) Balance is $6624      40
Agent DT1 deposits $314          (+) Balance is $6938      41
Agent WT1 withdraws $27          (-) Balance is $6911      42
Agent DT0 deposits $52           (+) Balance is $6963      43
Agent DT4 deposits $82           (+) Balance is $7045      44
Agent WT7 withdraws $73          (-) Balance is $6972      45
Agent DT3 deposits $143          (+) Balance is $7115      46
Agent DT2 deposits $35           (+) Balance is $7150      47
Agent DT1 deposits $346          (+) Balance is $7496      48
Agent DT0 deposits $275          (+) Balance is $7771      49
Agent DT4 deposits $415          (+) Balance is $8186      50

*** Flagged Transaction - Depositor Agent DT4 Made A Deposit In Excess Of $350.00 USD - See Flagged Transaction Log.

Agent DT3 deposits $379          (+) Balance is $8565      51

*** Flagged Transaction - Depositor Agent DT3 Made A Deposit In Excess Of $350.00 USD - See Flagged Transaction Log.

Agent DT2 deposits $11           (+) Balance is $8576      52
Agent WT4 withdraws $49          (-) Balance is $8527      53
Agent DT1 deposits $419          (+) Balance is $8946      54

*** Flagged Transaction - Depositor Agent DT1 Made A Deposit In Excess Of $350.00 USD - See Flagged Transaction Log.

Agent DT0 deposits $233          (+) Balance is $9179      55
Agent DT4 deposits $318          (+) Balance is $9497      56
Agent DT3 deposits $26           (+) Balance is $9523      57
Agent DT2 deposits $77           (+) Balance is $9600      58
Agent WT1 withdraws $60          (-) Balance is $9540      59
Agent WT9 withdraws $28          (-) Balance is $9512      60
Agent DT1 deposits $84           (+) Balance is $9596      61
Agent DT0 deposits $234          (+) Balance is $9830      62
Agent DT3 deposits $93           (+) Balance is $9923      63
Agent DT4 deposits $59           (+) Balance is $9982      64
Agent WT8 withdraws $53          (-) Balance is $9929      65
Agent WT3 withdraws $48          (-) Balance is $9881      66
Agent WT8 withdraws $17          (-) Balance is $9864      67
Agent DT2 deposits $139          (+) Balance is $10003      68
Agent DT1 deposits $272          (+) Balance is $10275      69
Agent DT0 deposits $142           (+) Balance is $10417      70
Agent DT3 deposits $438          (+) Balance is $10855      71

*** Flagged Transaction - Depositor Agent DT3 Made A Deposit In Excess Of $350.00 USD - See Flagged Transaction Log.

Agent DT4 deposits $395          (+) Balance is $11250      72

*** Flagged Transaction - Depositor Agent DT4 Made A Deposit In Excess Of $350.00 USD - See Flagged Transaction Log.

Agent WT6 withdraws $20          (-) Balance is $11230      73
Agent DT2 deposits $259          (+) Balance is $11489      74
Agent DT1 deposits $485          (+) Balance is $11974      75

*** Flagged Transaction - Depositor Agent DT1 Made A Deposit In Excess Of $350.00 USD - See Flagged Transaction Log.

Agent WT4 withdraws $55          (-) Balance is $11919      76
Agent WT6 withdraws $14          (-) Balance is $11905      77
Agent DT0 deposits $493          (+) Balance is $12398      78

*** Flagged Transaction - Depositor Agent DT0 Made A Deposit In Excess Of $350.00 USD - See Flagged Transaction Log.

Agent DT3 deposits $38           (+) Balance is $12436      79
Agent DT4 deposits $496           (+) Balance is $12932      80

```