# Sample Solution Template for A4

# Name: Rishabh Singh

# Student ID: 11748219

# Assignment details: A4 Debugging

# A4 Bug Report (https://github.com/RishCSU/ITC205\_Assessment4)

## Bug 1: Incorrect calculation of fines.

Bug 1 Solution (see end)

Replication:

# Scenario 1: Calculate Fine

## Scenario Description

* This scenario tests the fines that are incurred against the Patron when they attempt to pay the fine after returning overdue loan items.

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date | Author | Description |
| 1.0 | 21/10/2022 | Rishabh | Initial (and Final) Version |

## Test Scripts

The following scripts will cover this scenario:

* 1.1 Overdue Loan Case

## Use Case

* Paying an overdue fine.

## Test Components/Requirements

This test scenario covers the following high-level test requirements (see scripts below for specific requirements covered by each test script):

* Functioning of the

## User Groups

* The administrators and users of the BRLS.

## Script 1: Overdue Loan Case

### Script Description

* Overdue fine is calculated correctly

### Testing Requirements

This test script covers the following specific testing requirements:

* A member’s loan repayment amount should still be the same.

### Setup

* Patron created
* Item created
* Item borrowed

### Teardown

* Quit app

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Add a new Patron | New patron added | Pass |
| 2 | Add a new Item | New item added | Pass |
| 3 | Borrow Item | Patron 1 borrows item 1 | Pass |
| 4 | Increment day by 4 | Date increments 4 days | Pass |
| 5 | Return item | Patron 1 return item 2 two days late | Pass |
| 6 | Pay Fine | Patron 1 pays $2.00 fine | Fail |

### Test Data

|  |  |
| --- | --- |
| Step | Data |
| 1 | Current date: 21/10/2022  Patron Id: 1  Patron Name: Test Person  Patron email: [test@example.com](mailto:test@example.com)  Patron phone: 5551234 |
| 2 | Item Id: 1  Item type: Book  Item author: Another Person |
| 3 | Loan Id: 1 Loan due date: 23/10/2022 |
| 4 | Current date: 25/10/2022 |
| 5 | Patron 1 fines: $2.00 |
| 6 | Patron 1 fines: $6.00 |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date/Time | Tester | Test ID | Test Phase | Status |
| 21/10/22 11:35 am | Rishabh Singh | rsing107 | System Cycle 1 | Failed |

Step 1 Result: Pass

Adding patron functions normally



Step 2 Result: Pass

Adding item functions normally

Text

Description automatically generated

Step 3 Result: Pass – Borrowing one item functions normally

Text

Description automatically generated

Step 5 Result: Pass – When returning overdue book, fines are imposed as per normal.

Text

Description automatically generated

Step 6 Result: Fail

However, patron is charged incorrectly when paying fine.



Version control: Provide screenshots of commits

Text

Description automatically generated with medium confidence

**Simplification**

**H0: Fine amount is inaccurate when viewed at the Pay Fine System**

**T0: Set breakpoint at cardSwiped() in PayFineControl and see value of fines.**

**R0: Correct – the value of fines is $6.00, when it should be $2.00**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Now, backwards chaining to the Return Item System**

**H1: Fine amount is erroneous at the start of the ReturnItemControl class**

**T1: Set breakpoint when currentLoan is retrieved from library.**

**R1: Incorrect – the fine is currently 0.**

**Text

Description automatically generated**

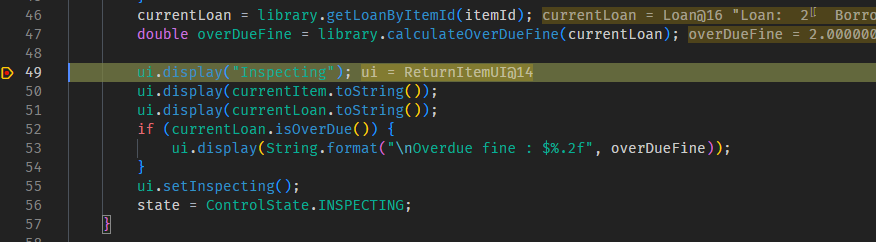
Text

Description automatically generated

H2: The library.calculateOverDueFine(itemId) method is inaccurate.

T2: Set a breakpoint to see the value of overDueFine.

R2: Incorrect – the overDueFine was calculated correctly.



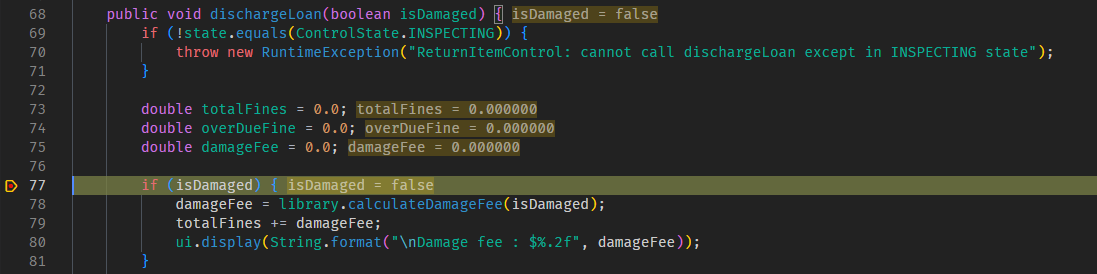
Graphical user interface, text

Description automatically generated

H3: isDamaged is accidentally true, adding additional fines

T3: Set a breakpoint in dischargeLoan() to check the value of isDamaged.

R3: Incorrect – isDamaged is false and if statement to add damagedFine is not triggered.



Graphical user interface, text, application, chat or text message

Description automatically generated

H4: currentLoan.getFines() calculates the fines incorrectly.

T4: Set a breakpoint to check value of currentLoans.getFines().

R4: Incorrect – the calculated fine is $2 as expected.

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

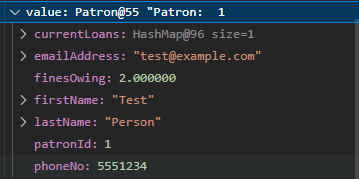
H5: patron.incurFine() has an error, and adds the wrong fine.

T5: Set a breakpoint to check the value of finesOwing after incurFines() is called.

R5: Incorrect – finesOwing is $2 after incurFines is called.

Text

Description automatically generated



H6: library.dischargeLoan()adds further fines on top of incurFines()

T6: Set a breakpoint to check value of finesOwing after dischargeLoan()

R6: Correct! finesOwing is now $6.

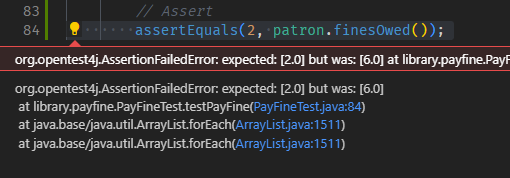
Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Method call narrowed down to library.dischargeLoans()



Wrote a unit test – failed as expected.

Text

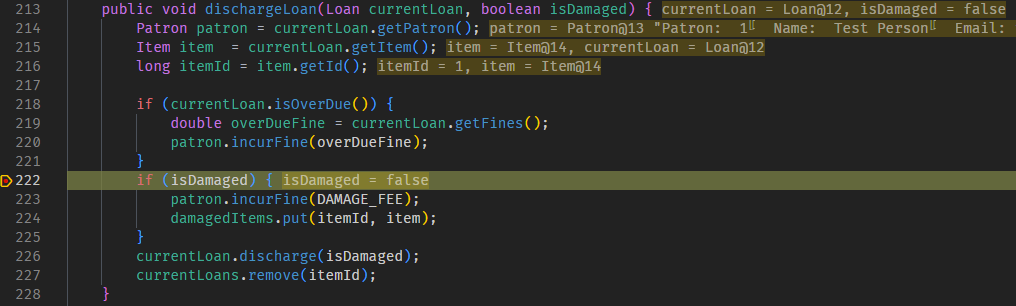
Description automatically generated with medium confidence

**Tracing**

**H0: patron.incurFine() in the library.dischargeLoan() method adds extra fine.**

**T0: Set a breakpoint immediately after the line of code, to see the value.**

**R0: Correct! finesOwing changes from 2 to 4. Part of bug found.**

****

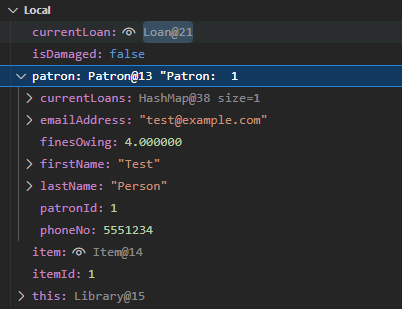
**Text

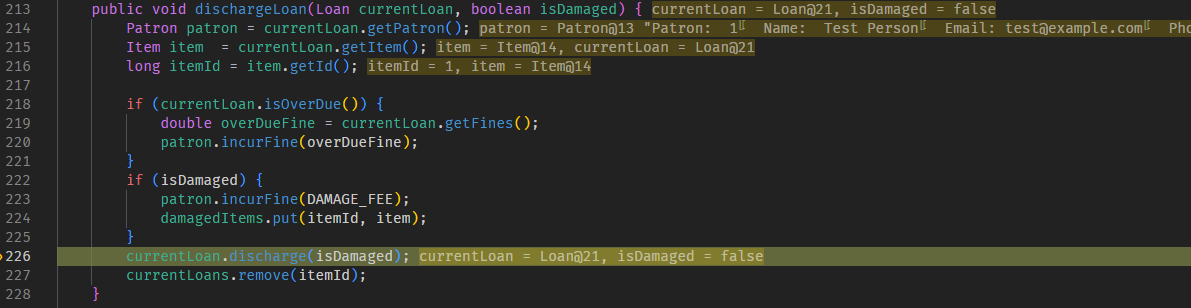
Description automatically generated**

**H1: isDamaged value is true, patron incurs extra fines.**

**T1: Set a breakpoint in the body of the if statement to see value.**

**R1: Incorrect, isDamaged is false.**

****

****

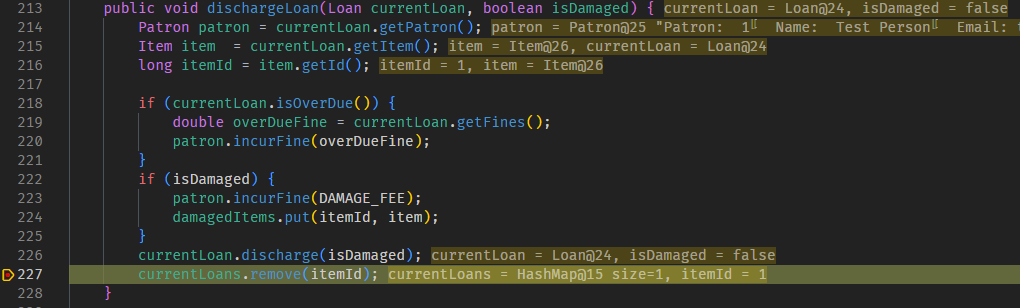
**H2: Error lies in currentLoan.dischargeLoan().**

**T2: Set breakpoint after currentLoan.dischargeLoan to check value of finesOwing.**

**R2: Correct! finesOwing is now $6.**

**Text

Description automatically generated**

****

**H3: The problem lies in patron.dischargeLoan() in the discharge() method.**

**T3: Set a breakpoint to compare value of variable after patron.dischargeLoan()**

**R3: Correct! Value of finesOwed is now 6.**

**Text

Description automatically generated**

**Text

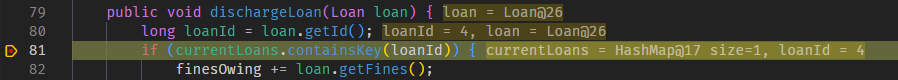
Description automatically generated**

**H4: The problem lies in loan.getID()**

**T4: Set a breakpoint.**

**R4: No. finesOwing is still $4.**

**Text

Description automatically generated** ****

**H5: The problem lies in finesOwing += loan.getFines()**

**T5: Set a breakpoint.**

**R5: Yes. finesOwing is now $6.**

**Text

Description automatically generated**

**Text

Description automatically generated**

Graphical user interface, text

Description automatically generated

**Resolution**

**H0: Commenting out finesOwing += loan.getFines() and patron.incurFine() will fix problem.**

**T0: Commented out said lines**

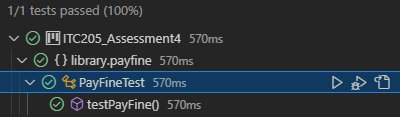
**R0: Yes! It fixes it. (when ran individually and without library.obj)**

**Graphical user interface, text

Description automatically generated**

**Text

Description automatically generated**

****

Text

Description automatically generated with low confidence

## Bug 2: Patron can borrow more than LOAN\_LIMIT

Bug 2 Solution

Replication:

# Scenario 2: Calculate Fine

## Scenario Description

* Borrowing more than the allowed amount of items.

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date | Author | Description |
| 1.0 | 22/10/2022 | Rishabh Singh | Initial and Final Version |

## Test Scripts

The following scripts will cover this scenario:

* 1.1 Borrowing more than 2 items.

## Use Case

* Borrowing an appropriate and legal amount of items.

## Test Components/Requirements

This test scenario covers the following high-level test requirements (see scripts below for specific requirements covered by each test script):

* Functioning BRLS components

## User Groups

* The administrators and users of the BRLS.

## Script 2: Overdue Loan Case

### Script Description

* An attempt is made is to borrow more than the allowed amount of items

### Testing Requirements

This test script covers the following specific testing requirements:

* Patrons cannot borrow

### Setup

* Patrons and items are added
* Loan is taken out

### Teardown

* Program is quit

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Add a new Patron | New patron added | Pass |
| 2 | Add 4 new Items | 4 items added | Pass |
| 3 | Borrow Item | Patron 1 borrows item 1 | Pass |
| 4 | Borrow another item | Patron 1 borrows item 2 | Pass |
| 5 | Borrow another item | Not allowed as loan limit reached | Fail |
| 6 | Enter Borrow system again | Not allowed as Patron is not allowed | Fail. |

### Test Data

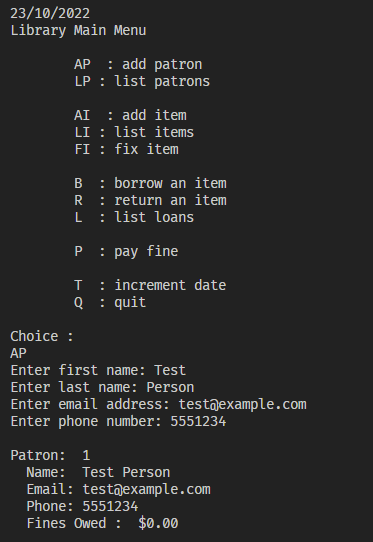
|  |  |
| --- | --- |
| Step | Data |
| 1 | Current date: 21/10/2022  Patron Id: 1  Patron Name: Test Person  Patron email: [test@example.com](mailto:test@example.com)  Patron phone: 5551234 |
| 2 | 4 new items. |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date/Time | Tester | Test ID | Test Phase | Status |
| 22/10/22 11:35 am | Rishabh | Rsing107 | System Cycle 1 | Failed |

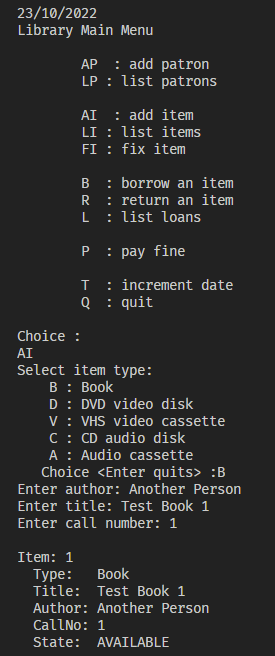
Step 1 Result: Pass

Adding Patron functions normally.



Step 2 Result: Pass

Adding 4 items functions normally.



A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Step 3 Result: Pass

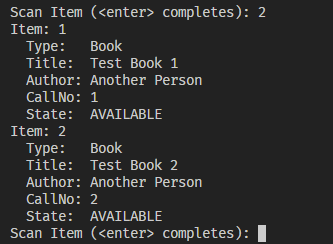
Borrowing first item functions as normal

Text

Description automatically generated

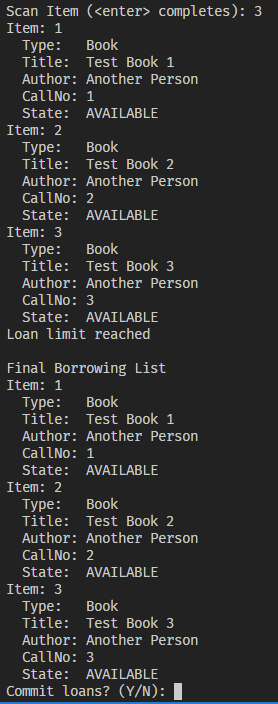
Step 4: Pass

Borrowing second item also functions as normal



Step 5: Fail

Third item should not be able to be borrowed.



Step 6: Fail

Entering the Borrow Item Use Case to borrow a fourth item should not be allowed.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

and performed

**Simplification**

**H0: Error lies in commitLoans()**

**T0: Set a breakpoint.**

**R0: No, commitLoans() is not called when third item placed. Breakpoint not triggered.**

**H1: Error lies in borrowingComplete()**

**T1: Set a breakpoint.**

**R1: No, borrowingComplete() not called until after third item. Breakpoint not triggered**

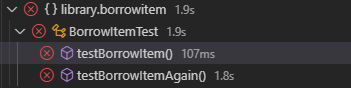
**H2: Error lies in itemScanned()  
T2: Set a breakpoint.  
R2: Yes. Breakpoint triggered on all items. Doesn’t stop at second one.**

***Difficult to show screenshots for these.***

Graphical user interface, text, application

Description automatically generated

**Wrote unit tests – these failed.**

****

**Text

Description automatically generated with medium confidence**

**Text

Description automatically generated**

**Text

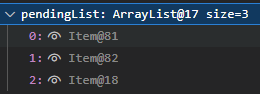
Description automatically generated**

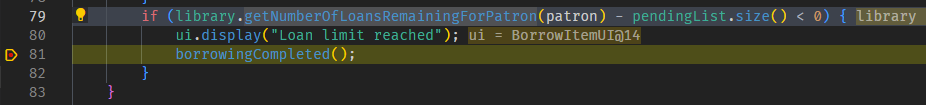
**Tracing**

**H0: pendingList.size() or getNumberOfLoansRemainingForPatron() are wrong.**

**T0: Breakpoint to test.**

**R0: No, they are accurate.**

****

****

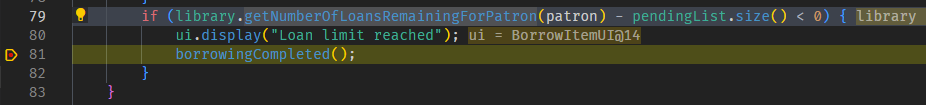
**H1: Comparison is working correctly.**

**T1: Compare values at breakpoint**

**R1: Yes, doesn’t activate when pendingList.size() is 2, when it should.**

**Graphical user interface, text, application

Description automatically generated**

****

Graphical user interface, text

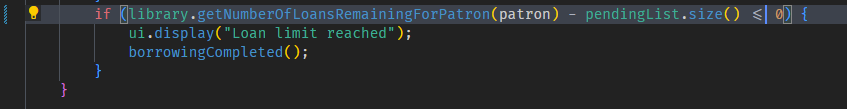
Description automatically generated

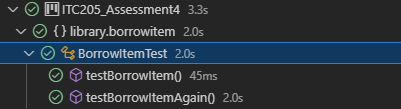
**Resolution**

**H0: Changing comparison to <= will solve the problem.**

**T0: Changed**

**R0: Tests now pass!! (when ran individually and without library.obj)**

****

****

**Graphical user interface, text

Description automatically generated**