# Sample Solution Template for A4

# Name: Rishabh Singh

# Student ID: 11748219

# Assignment details: A4 Debugging

# A4 Bug Report

## Bug 1: Incorrect calculation of fines.

Bug 1 Solution

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

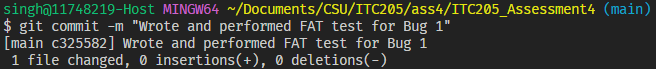
Description automatically generated

Step 6 Result: Fail

However, patron is charged incorrectly when paying fine.



Version control: Provide screenshots of commits

****

**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**

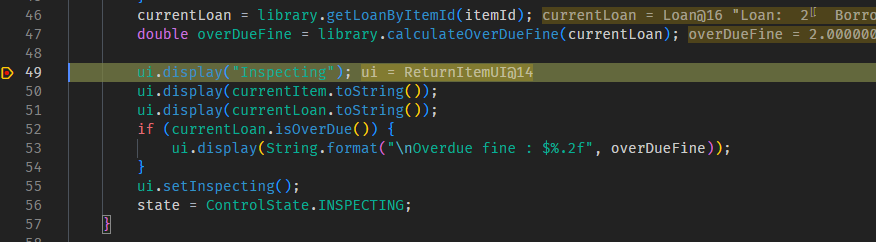
Text

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.



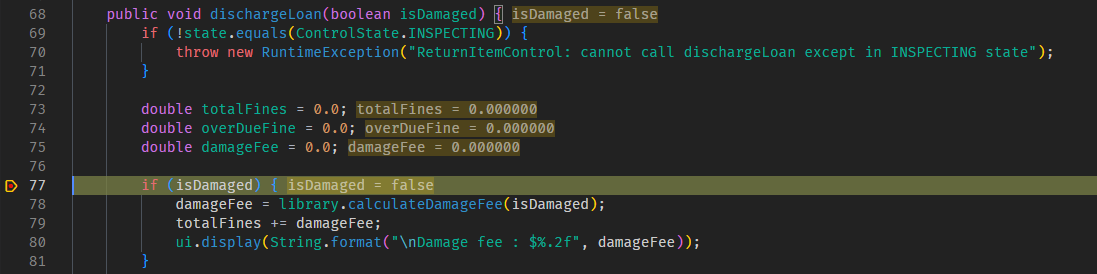
Graphical user interface, text

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

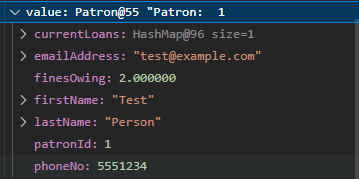
H5: patron.incurFine() has an error, and adds the wrong fine.

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.

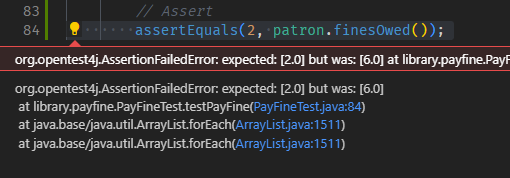
Text

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.

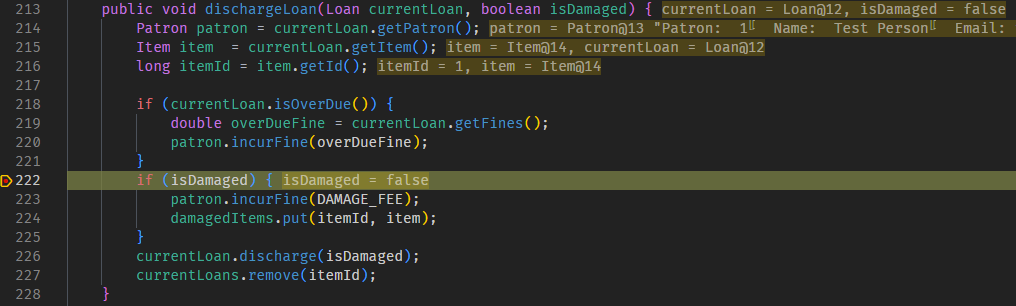
Version control: Provide screenshots of commits

**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! finesOwed changes from 2 to 4. Part of bug found.**

****

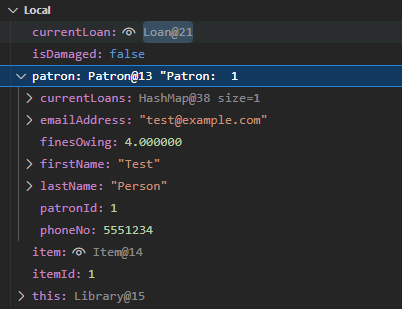
**Text

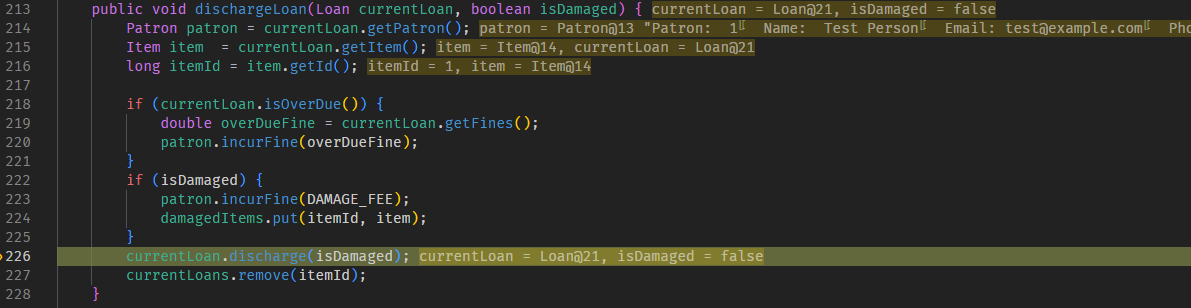
Description automatically generated**

**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.**

****

****

Version control: Provide screenshots of commits

**Resolution**

**H0:**

**T0:**

**R0:**

**Test screenshots with correct code**

Version control: Provide screenshots of commits

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

Bug 2 Solution

Replication:

# Scenario 2: Calculate Fine

## Scenario Description

* Describe the intent of the test scenario
* Test scenarios (aka test set or test suite) are a set of test scripts that cover a specific functional area, business process, use case, etc…
* For example, a given test scenario might cover an order placed on a web site by an existing customer (another scenario might cover orders placed by new customers) – test scripts within the scenario might cover a single item order, a multiple items order, quantity not on hand exception, etc…

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date | Author | Description |
| 0.1 | 01/01/2006 | Author Name | Initial Draft |
| 1.0 | 01/10/2006 | Author Name | Initial Version |

## Test Scripts

The following scripts will cover this scenario:

* 1.1 Overdue Loan Case
* 1.2 Test Script Name
* 1.3 Test Script Name

## Use Case

* List the Use Case covered by this Test Scenario – limit the test scenario to just one Use Case

## Test Components/Requirements

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

* Component 1
* Component 2
* Functional Requirements Group 1

## User Groups

* User group/responsibility 1
* User group/responsibility 2

## Script 2: Overdue Loan Case

### Script Description

* Overdue fine is calculated correctly

### Testing Requirements

This test script covers the following specific testing requirements:

* A member cannot borrow when they have an overdue loan.

### Setup

* Member has an overdue loan
* The book returned after the due date

### Teardown

* List all steps that should be taken after the test case is executed

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | The patron chooses the option r | Calculate 1 day fine | Pass/fail |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |

### Test Data

|  |  |
| --- | --- |
| Step | Data |
| 1 | Member Id: 1 |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date/Time | Tester | Test ID | Test Phase | Status |
| 2/12/06 11:35 am | Joseph Jones | Samsmith1 | System Cycle 1 | Failed |
| 2/16/06 3:45 pm | Joseph Jones | Samsmith1 | System Cycle 2 | Passed |

Step 1 Result: Pass

<screenshots>

Version control: Provide screenshots of commits

**Simplification**

**H0:**

**T0:**

**R0:**

**Test screenshots of the buggy code**

Version control: Provide screenshots of commits

**Tracing**

**H0:**

**T0:**

**R0:**

**Test screenshots of the buggy code**

**Description of the screenshots**

Version control: Provide screenshots of commits

**Resolution**

**H0:**

**T0:**

**R0:**

**Test screenshots with correct code**

Version control: Provide screenshots of commits