# UAT Bug 1

# Name: Mina

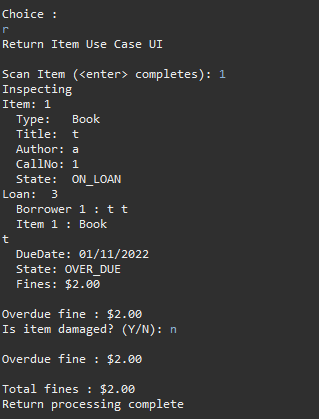
# Student ID: 11716113

# Assignment details: Debugging

# A4 Bug Report

## Bug 1: Incorrect calculation of fines.

Excessive fines are applied when the patron tries to pay fine.



Text

Description automatically generated

Bug 1 Solution

**Replication:** The fines for an overdue item are being applied and increased by a method that discharges the loan.

# Scenario 1: Over due Loan payment

## Scenario Description

* This scenario is made to test if the fines applied remain consistent throughout the return and payment process.
* The test is used to determine which part of the program is faulty and needs adjustment in order to function as intended
* This scenario is based on a patron’s need to return an over due item and pay the over due fees.

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date | Author | Description |
| 0.1 | 18/10/2022 | Mina Farag | Initial Draft |
| 1.0 | 18/10/2022 | Mina Farag | Initial Version |

## Test Scripts

The following scripts will cover this scenario:

* 1.1 Overdue Loan Case
* 1.2 Return item case
* 1.3 Paying fine case

## Use Case

* The use case covered by this scenario will be discharge of loan.

## Test Components/Requirements

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

* Patron
* Loan
* Item

## User Groups

* Actor/Patron

## Script 1: Overdue Loan Case

### Script Description

* Overdue fine is calculated incorrectly

### Testing Requirements

This test script covers the following specific testing requirements:

* A member cannot borrow when they have an overdue loan.
* A member must be charged the correct amount for having an overdue loan

### Setup

* Member has an overdue loan
* A member’s details must be input into the system
* The loan data must be input into the system
* An item must be input into the system
* The item returned after the due date

### Teardown

* Attempt to fix issue of incorrect loan charge

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Patron attempts to return a previously borrowed item | Item condition must be inspected and results must increase or not affect total fine | Pass |
| 2 | Patron attempts to discharge overdue loan | Loan fines owed present the correct fine and correct payment is accepted | Fail |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |

### Test Data

|  |  |
| --- | --- |
| Step | Data |
| 1 | Member Id: 1 |
| 2 | Item Id: 1 |
| 3 | Loan Id: 1 |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date/Time | Tester | Test ID | Test Phase | Status |
| 3/11/2022 | JJ | 1 | Step 1 | Passed |
| 3/11/2022 | JJ | 1 | Step 2 | Fail |

Step 1 Result: Pass

<screenshots>

Text

Description automatically generated

**Screen shot displays correct fine applied for over due item after 2 days over the due date.**

Step 2 Result: Fail

Text

Description automatically generated

Owed fines is multiplied by 3 which is the incorrect amount of fines that the patron should be charged.

Version control:

A picture containing graphical user interface

Description automatically generated

**Commit of UATB1 doc to Version Control Repo**

**Simplification**

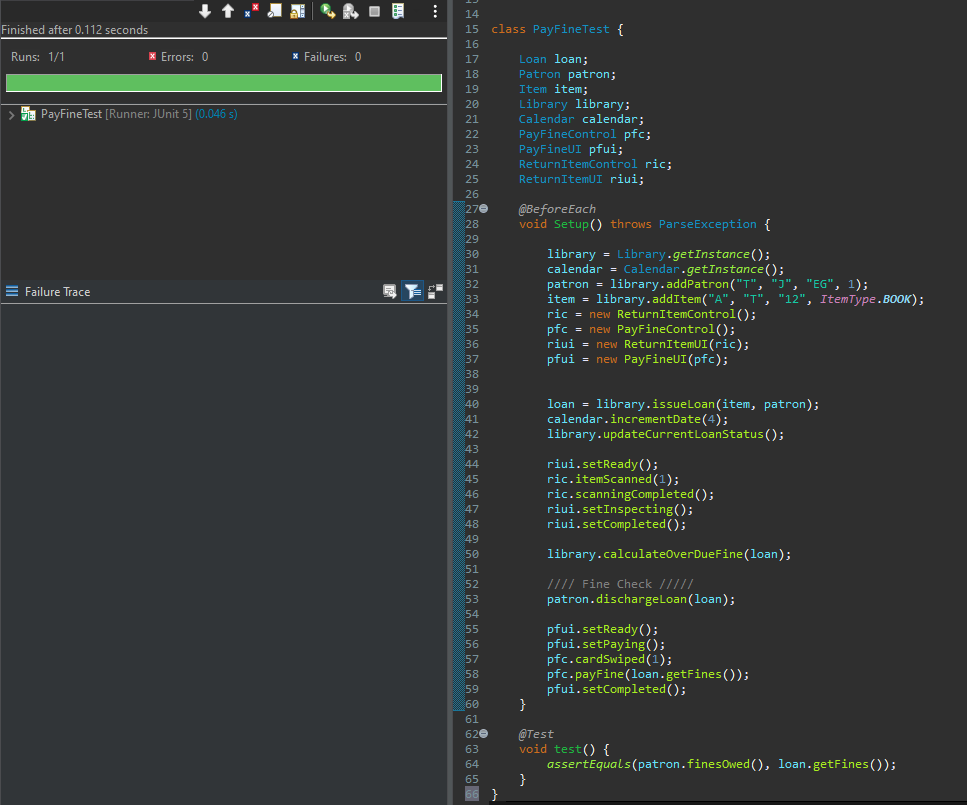
**H0: My guess is that the problem is to do with the issuing of fines and their calculation. Because the final fine returns the bug when attempting to pay, I am guessing that the problem resides within one of the methods that calculate the final fine or display it.**

**T0: I will be testing three methods being patron.dischargeLoan(), loan.discharge() and library.dischargeLoan().**

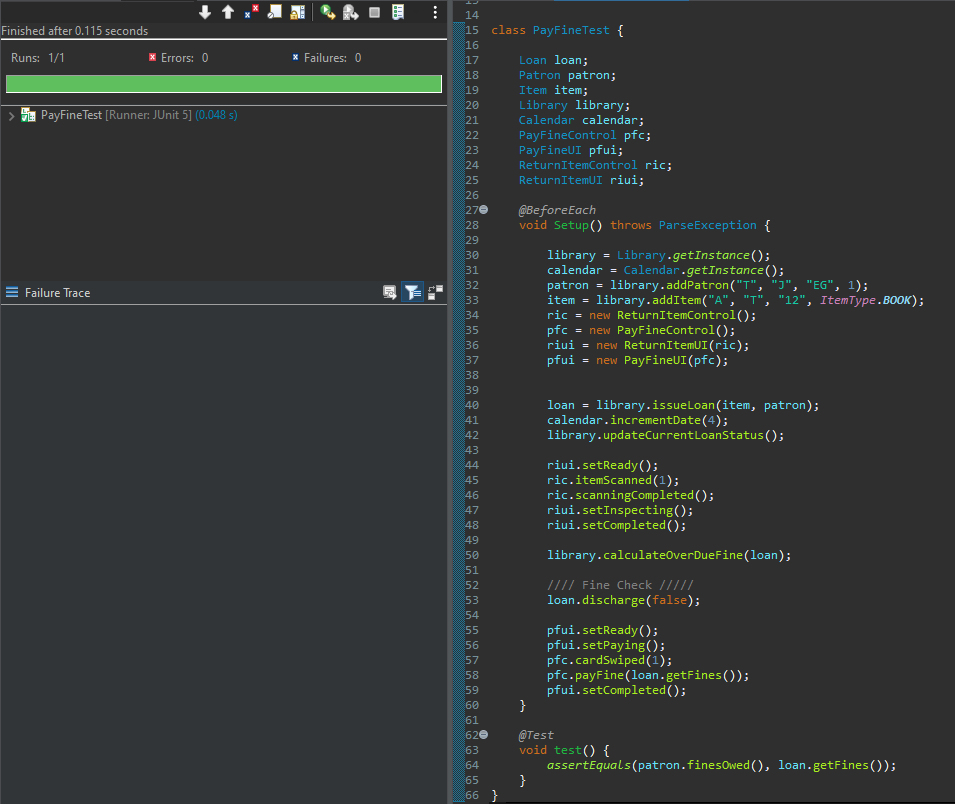
**R0: Below is the tests I have conducted on three methods that manage the loan output**

**(Screenshots)**

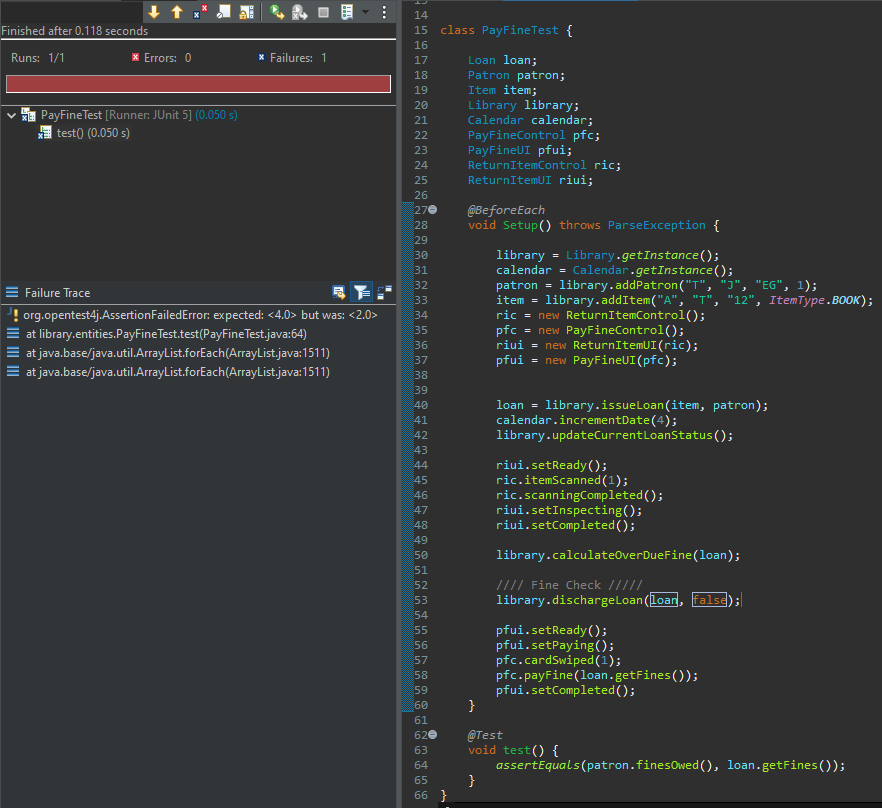
**Testing patron.dischargeLoan() method (Expected that patron.finesOwed() and loan.getFines() must both be equal to $2) PASS**



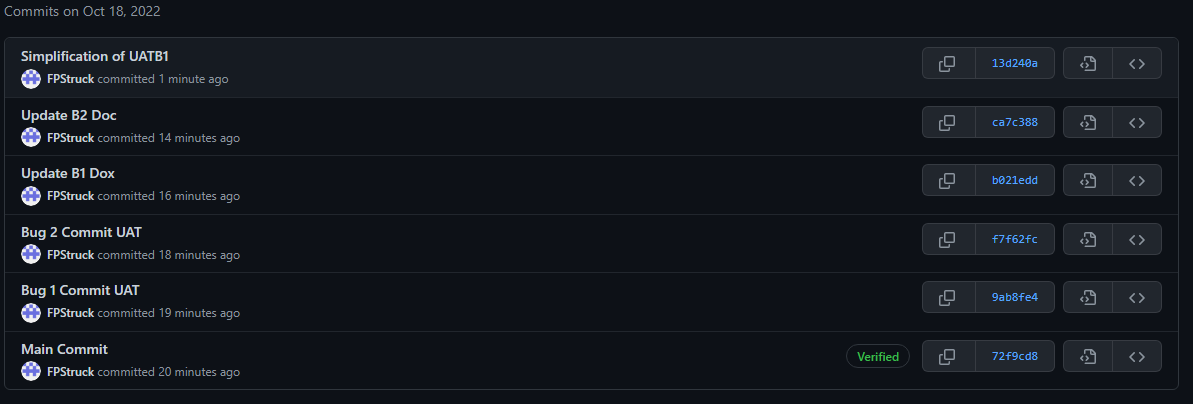
**Testing loan.discharge() method (Expected that patron.finesOwed() and loan.getFines() must both be equal to $2) PASS**



**Testing library.dischargeLoan() method (Expected that patron.finesOwed() and loan.getFines() must both be equal to $2) FAIL: Patron is charged $4 when they should only have to pay $2**



**Version control:**



**Commit of UATB1 doc to Version Control Repo**

**Tracing**

**The test conducted below is to make sure that the problem occurs before any changes are made to the code**

**Text

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence**

**H0: My hypothesis is that the bug occurs from a method that returns/calculates the final fine owed by a patron who wants to return an over due item and pay for it. I believe that this problem can occur in either the patron or library file. After inspecting the library file and the pay fine use case ui and control files. I have determined that the patron file is the reason for the current error calculating fines.**

**T0: Tests have been conducted in the debug log. It is safe to say that the origin of the bug occurred in two variables within the patron java file. Below are the two variables that caused the bug.**

**R0:**

**Before origin is identified:**

Text

Description automatically generated

Sane Variables

**Result output of returning a single item (Undamaged):**

**A screenshot of a computer

Description automatically generated with medium confidence**

**After origin is identified:**

**Text

Description automatically generated**

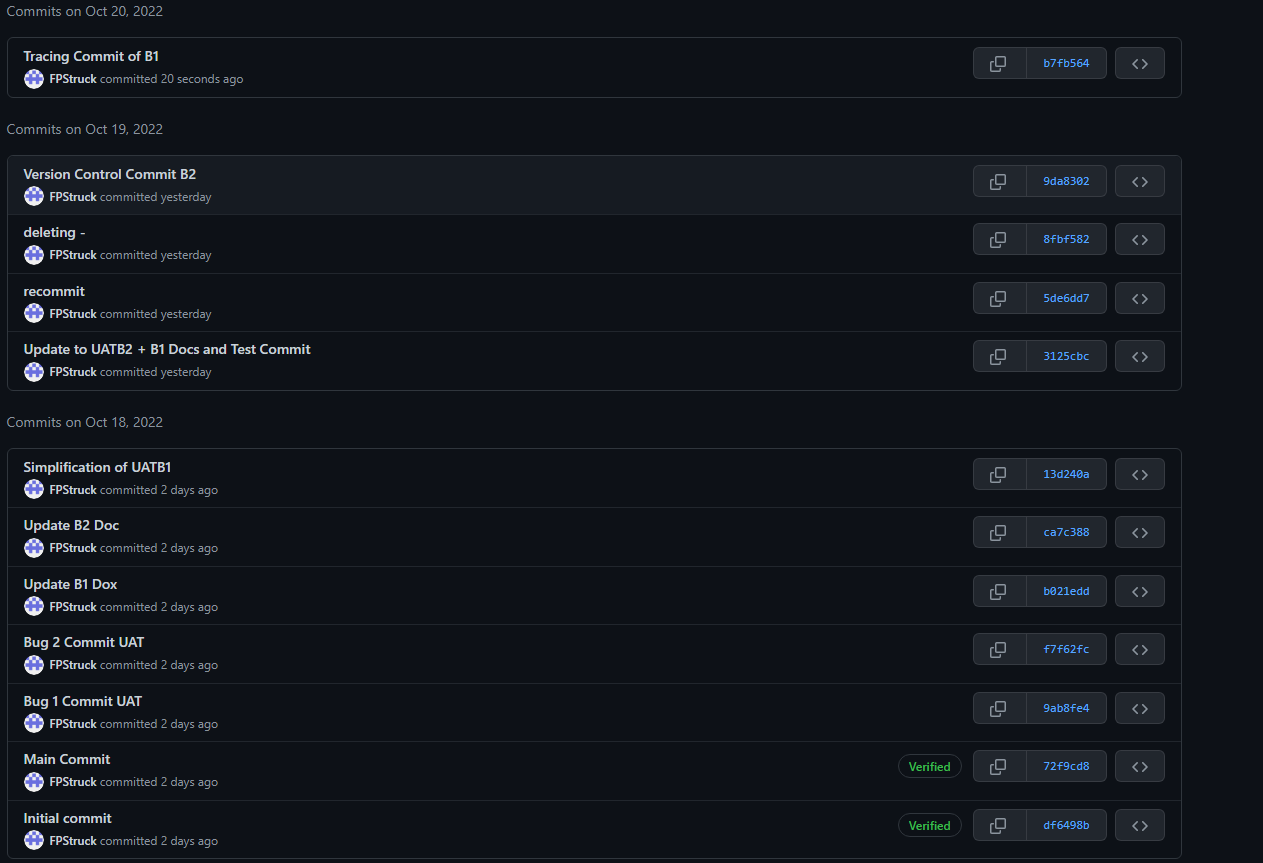
Infected code

Infected code

**Result output of returning a single item (Undamaged):**

Text

Description automatically generated

**Version control:** 

**Resolution**

**H0: After using the tracing method to determine the location of the infected code. The code has been modified and resolved as tested by the program. I believe that the JUNIT test will return the same accurate results after implementation.**

**T0: This test will be a repeat of the first JUNIT test to determine whether the code error has been resolved or not.**

**Previous test results before bug resolution:** **(The result should be 2)**Text

Description automatically generated

**Current test results before bug resolution: (The result is 2)**

**Text

Description automatically generated**

**R0: The bug has been successfully resolved**

**Correct code:**

**Text

Description automatically generated**

**Test screenshot:**

**Text

Description automatically generated**

**Version control:**



**User test case**

# Scenario 1: Over due Loan payment

## Scenario Description

* This scenario is made to test if the fines applied remain consistent throughout the return and payment process.
* The test is used to determine whether all buggy parts of the program have been resolved
* This scenario is based on a patron’s need to return an over due item and pay the over due fees. Includes a damaged item

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date | Author | Description |
| 0.1 | 20/10/2022 | Mina Farag | Initial Draft |
| 1.0 | 20/10/2022 | Mina Farag | Initial Version |

## Test Scripts

The following scripts will cover this scenario:

* 1.1 Overdue Loan Case
* 1.2 Return item case
* 1.3 Paying fine case

## Use Case

* The use case covered by this scenario will be discharge of loan.

## Test Components/Requirements

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

* Patron
* Loan
* Item

## User Groups

* Actor/Patron

## Script 1: Overdue Loan Case

### Script Description

* Overdue fine is calculated incorrectly

### Testing Requirements

This test script covers the following specific testing requirements:

* A member cannot borrow when they have an overdue loan.
* A member must be charged the correct amount for having an overdue loan

### Setup

* Member has an overdue loan
* A member’s details must be input into the system
* The loan data must be input into the system
* An item must be input into the system
* The item returned after the due date

### Teardown

* If error occurs. Attempt to fix. Otherwise, test was successful.

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Patron attempts to return a previously borrowed item | Item condition must be inspected and results must increase or not affect total fine | Pass |
| 2 | Patron attempts to return a previously borrowed item that is damaged | Item condition must be inspected and results must increase or not affect total fine (damaged) | Pass |
| 3 | Patron successfully returns both items and attempts to pay | Pay fine use case is called and displays correct output result | Pass |
| 4 | Patron pays correct amount | System ends pay fine use case | Pass |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |

### Test Data

|  |  |
| --- | --- |
| Step | Data |
| 1 | Member Id: 1 |
| 2 | Item Id: 1, 2 |
| 3 | Loan Id: 1, 2 |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date/Time | Tester | Test ID | Test Phase | Status |
| 24/10/2022 | JJ | 1 | Step 1 | Passed |
| 24/10/2022 | JJ | 1 | Step 2 | Passed |
| 24/10/2022 | JJ | 1 | Step 3 | Passed |
| 24/10/2022 | JJ | 1 | Step 4 | Passed |

**Screenshots of scenario:**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**