Asg 4 Guidance

It is not sufficient simply to eliminate the reported bugs. You need to record the debugging process. This means that you need to record the sequence of hypotheses (guesses) and tests (checks) that lead you from the point of failure (where the bug manifests as output) BACKWARDS to the origin of the defect.

You should record each debug log in this form:

Bug X

Point of Failure: At line XX in main, <varY> is incorrect.

H0: var Y is reported incorrectly

T0: set a breakpoint at line XX, visually inspect varY value before output.

Result0: H0 false: varY is reported correctly.

H1: at line XY (some earlier line that affects the value of varY), varZ (some var used to affect varY) is incorrect.

T1: set a breakpoint at line XY, visually inspect varZ value

Result1: H1 true: varZ is incorrect

//////////

You should record each debug log in this form:

Bug 1

Point of Failure: Potential error at line 226 in Library.java, Variable currentLoan.discharge(isDamaged) is incorrect. Removed line.

H0: currentLoan.discharge(isDamaged) is reported incorrectly

Text

Description automatically generated

T0: Variable may return the item is damaged even if it isn’t. Variable will be removed and program will be tested for results.

Result0: H0 false: the variable is reported correctly, other methods may be interrupting correct output a potential file that should be investigated should be the patron file, further errors occurred during running of program from H0. Item 3 is constantly in loan state and cannot be borrowed. Library file is not the source of infected code. Restoring Library.java file to original state.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Point of Failure: Potential error at line 82 and 102 in Patron.java, Variable finesOwing += loan.getFines() and finesOwing += fines may be incorrect. Modified line 102 to finesOwing += fines – finesOwing and line 82 to finesOwing = finesOwing + loan.getFines().

H1: owed fines are reported incorrectly, the result of modification should present correct results based on a series of tests for both damaged and undamaged items with total fines obtained.

**Before**

Text

Description automatically generated

**After**

Text

Description automatically generated

T1: Inspection of both line 82 and line 102 variables will be inspected and tested

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Result1: H1 true: Variable 82 and 102 play a role in the bug. They both were infected but the incorrect result has been output. Further modification to these variables required.

Point of Failure: Modification of line 102 and 82 in the patron file should result in a correct output

H2: Refinement to fines owed calculations in patron file should result in correct output. Modifications to line 102 were made from finesOwing += fine – finesOwing to finesOwing += fine – 2. Line 82 will stay the same to investigate if line 102 was the issue

**Before**

Text

Description automatically generated

**After**

Text

Description automatically generated

T2: Modifications to line 102 will be tested for correct output. Testing one normal item and one damaged item for correct output result. Then for the final test I will test both normal and damaged together.

**Normal Item (undamaged) pay fine test:**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generatedText

Description automatically generated

**Damaged item pay fine test:**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generatedText

Description automatically generated

**Damaged item and normal item (undamaged) pay fine test:**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generatedText

Description automatically generated Text

Description automatically generated

Result2: H1 true: Variable 82 and 102 play a role in the bug. Both variables were infected and resulted in incorrect outputs. The modification of these variables eliminated the bug and correct results were obtained.

Upon testing for two damaged items to confirm the bug is eliminated. This is the pay fine use case ui result

Text

Description automatically generatedText

Description automatically generated Text

Description automatically generated

(H0 etc represent your hypotheses (or guesses) and T0 etc represent the test (or check) you make to validate or falsify your hypotheses) Continue to record hypotheses and tests until you find the bug.

Note that you are not required to write a Junit test for each step – a test can be a prediction on what you will observe at a particular break point in a debugger.

You will need to modify the format suggested above to reflect the exact hypotheses and tests that you perform. The point is that you need to document the logical progression of your debugging process.

Remember – you need to either work back up the dependence chain from the point of failure (externally observable fault), or progressively narrow down the area of code where the bug could be located. Just jumping straight to where you think the bug must be will not get you maximum marks. Maximum marks are for demonstrating you understand the process of scientific debugging and the steps involved in simplifying, isolating, testing, and rectifying, the bug then validating your fix.

If you are looking for higher grades (DI and HD) you should also write an automated test using JUnit (and Mockito as appropriate) that simplifies and replicates the bug. It is usually best to make the bug test fail when the bug is expressed and for the test to pass when correct behaviour is demonstrated.

Simplify where you think the bug should be down to a single function call (perhaps a pretty high level function’, set up the initial conditions for the bug to be expressed (perhaps by setting up some patrons/items and loans, then incrementing the date as appropriate)