

W04 Ponder: White Box Testing

Group 7

Bug report Template

Title. I.E. function x isn't named according to standard conventions

Steps To Reproduce

1. In the file x, go to line y
2. This function is named x but does not perform the function as expected
3. This function should be renamed to z

Code: FileName.cpp:Line X

Bug Reports

Inconsistency Errors (Ryan)

Division will error when divisor variable is set to 0

Steps to Reproduce

1. Execute the getRoman function
2. When the divisor variable get to 0, it will cause the division on the next line to fail
3. This will produce inconsistent results

Code: RomanNumerals.cpp:Line 63

The code base looks very consistent in syntax, naming, etc.

Inefficiency Errors (Tony)

Too many loops when translating roman to decimal

Steps to Reproduce

1. Execute the getDecimal function
2. Note that the operation of it is three times longer than if we loop once. It is possible turning the for loop into a do..while.

Code: int getDecimal (string input): Line 26

The “quit” option (see ambiguity Errors) transformed unnecessarily .

Steps to Reproduce

1. Starting the program should start the main method.
2. The user should be advised what to type to quit.
3. Change the line to `cout << "Enter a Roman numeral or type "q" to quit\n";`

Code: RomanNumerals.cpp:Line 84

Ambiguity Errors (Jacob)

The user is prompted to enter a roman numeral or quit.

Steps to Reproduce

4. Starting the program should start the main method.
5. The user should be advised what to type to quit.
6. Change the line to `cout << "Enter a Roman numeral or type "quit" to quit\n";`

Code: RomanNumerals.cpp:Line 84

Add `{ }` after each **if** statement for readability

1. Line 22
2. Line 43
3. Line 88

Add comments on each function for clarity and readability

1. Throughout

Narratives

Jacob Thomas

We spoke via slack about potential problems as we went. This helped me as I had forgotten a bit about the syntax used. Individually, I went through the entire program manually starting and ending with the main method while I combed for anything ambiguous.

Ryan Dockstader

I started with examining the code base to make sure syntactically everything looked correct, and consistent. I looked specifically for naming conventions in variables and functions, casings, and

spacing to make sure the code was readable and consistent. I then started to look at each function individually and test in my mind how it would behave with various inputs (integers, strings, null values, negative values, etc.) Then We all discussed various potential issues in the code via a slack conversation

Nick Campbell

Because I don't check my email as often as I should, I hadn't set up Slack yet, so I missed out on the initial communication on the assignment before I knew we had started communication. The first, and most obvious issue I found on the code was the lack of comments, needlessly increasing the difficulty to read the code. It didn't even mention the desired output of the program. I would personally include more spacing between sections, as the current code only separates declared variables from the rest of the function. Still, we managed to determine the code's meaning enough to find several errors.

Tony Moraes

The communication was an issue itself because I did not read the announcement about the slack group. After that I began to actively participate in the discussions giving resources and ideas to get the work done. From now on it is possible to check the code in a versioning control environment (github) that it is a great tool for further interaction of the team.

Reading the code we could not find extraordinary bugs or issues that could compromise the functionality. Some of them we are not sure if could be applied, so it was not added in this document. After all, the group was very interested and compromised reviewing the code presented.