

# Assignment 3

## Roman Numerical Converter

### Reengineering in Cobol

### Reflection Report

#### Programming Process/Reflection:

I do not like coding in Cobol. The paradigms of the language are very different from what I am used to and have experience with. Even though the syntax is “English like” I found it very confusing and verbose. My biggest challenge was understanding the algorithm in the code and how to properly modernize and reengineer the application. I would also like to mention that I was not able to properly install the compiler. Despite hours of trial and error I had to avail. In the end I just ended up using git and running it on the school server.

The bug that was in the code was easy to detect but unfortunately that is where the simplicity ended. Due to the convoluting working-storage section I found it very hard to diagnose where the issue is actually being initiated. Once the root of the problem was highlighted solving it became a trivial mathematical analysis. It is important to note that the complexity of this bug is not from the technical issue of the bug itself but instead the confusing nature of Cobol’s language itself. I would like to point out that just like in any other language it was very satisfying to solve the bug.

I have a friend who is about to start a co-op position where his primary role will be working on Cobol programs that are thousands of lines long. I have no envy for my friend. Because of its structure style and lack of readability the longer and more complicated the program becomes the exponentially harder it is to re-engineer it. Cobol is a strong language in its functionality and reliability but lacks in its readability, portability, and ability be extended.

With all that being said, I still understand why Cobol has survived as long as it has. This is due to the fact of Cobol’s robust file I/O specifications. The ability to specify exactly the content type of input/output files can be leveraged to a high degree, increasing the efficiency of the overall program.