Ebad Babar

0954164

CIS 3190 A3 Reflection

Working with Cobol

Reflection

Working with Cobol was certainly a challenge despite already have working knowledge of other programming language. Cobol is unlike anything else. For a while I was treating it like C and tried to manipulate string as I would have in other languages. Even the syntax was hard to get use to. Even after having done this assignment, I'm not feeling 100% in my abilities to work with Cobol.

Cobol was made challenging due to it's unique style. The most difficult part of the entire assignment was to get used the new style of programming and wrapping my head around how to use the language to do what I want. Also, the lack of function was really annoying, as well as the loops being replaced by perform.

What I enjoyed about Cobol was how similar it felt to writing Sudo code. Also, after understanding the syntax the I liked how easy the manipulation of the data was. Commands such as tallying and being able to write in English made programming fun and enjoyable.

"Cobol was designed to be complex" is what I disliked about the language the most. Despite its usability and functionality overall, I did not enjoy it. I spent 3x or 4x longer trying to understand the syntax and how the language worked then actually working on the assignment.

Code

How to compile:

```
code -x -free romanA3_1.cob conv.cob
```

How to run:

```
./romanA3_1
```

Conv.cob

I didn't change much in the conv from the original file that was given.

Changed all the go to statements to if-else

Added q to allow the user to exit the program after selecting n

Added invalid input catch

Move the adding to a function / new paragraph instead of putting in every if statement and removing it from a go to

romanA3_1.cob

Added more declarations

Converted all the go to statements to perform functions

Added more options to allow the user to quit, pick file, or manually enter data.

File not working.

L1 sets everything to 0

L2 is a recursive function that that's the user data and puts it in a field

The field is originally filled with 35 spaces which are over written by the user data

I use tally to count the remaining spaces to the length of user data.

Send to be manipulated

Does recursively till done

String-manipulation takes the user data as "string" and treats it like a "char" sending data to cove one at a time

B1 Calls the conv and deals with the returns

B2 Displays the answers

B3 End the program