

Assignment 3: Cobol ISBN Validation

Drew Mainprize 1045298

For assignment 3, I created an ISBN number validator using Cobol. The program takes in a file with a list of ISBN numbers and checks them for multiple issues. The issues it checks for include invalid digits (alphabetical characters) and check digits that do not equal the check sum of the number. I used many different features of the Cobol language including paragraphs, sections, perform loops, string formatting, etc.

I found Cobol to be a fairly easy language to pick up. The syntax was very light and resembled the English language to the point where I found myself writing pseudocode that happened to work within my program. It was also easy to work with variables in Cobol as everything pretty much has global scope. This made it easy to create flags for different conditions that I could check and update from anywhere in my program, and be able to handle those flags in a different part of my program without having to think about it. This did lead to some issues however as I needed to have lots of variables instead of being able to reuse common index variables (like i, j, etc.).

Some of the aspects of Cobol that made it a more challenging language to get used to were perform loops. I found that it was somewhat difficult to make them work properly, especially when I needed to nest the loops to the point where I had separate paragraphs for the inner and outer loops at one point just to make the program run. I ended up fixing this as it was a nightmare for anyone who enjoys looking at clean code, but it did give me a few issues. I also would have liked to have access to user-defined functions for this assignment. The paragraphs were very nice to work with in terms of using global variables, however I feel like I used them exactly like I used functions and functions would have made my program a lot easier to read instead of the paragraphs that really just resemble go-to statements and labelling.

Overall, some of the things I liked about Cobol were paragraphs and the fact it was easy to read. Paragraphs were like functions except all the variables within them had global scope. This was extremely useful when I needed multiple variables (flags) and I didn't have to pass them all around as parameters or mess with scope too much. The fact that it was easy to read also made Cobol a very easy language to begin programming in. I didn't have to read too much into the documentation before I was writing programs and learning the ins and outs of the language. One thing I disliked was the overcomplicated syntax for computation. Having to write an entire sentence just to add one variable to another got tedious quickly and made me second guess myself constantly as it was unclear what each computational statement was doing right away from someone who is used to the traditional +, -, /, * operators.