Adam Walton   Project 2   CIS 4930 COBOL for FinTech
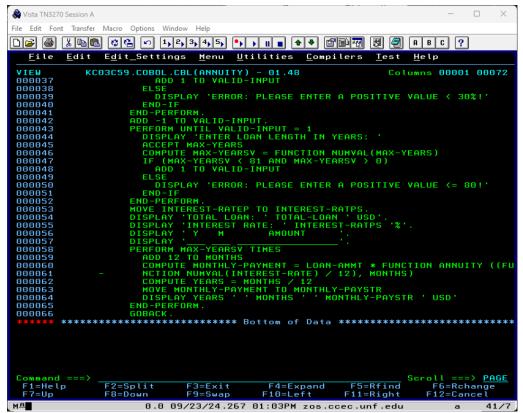
Screenshot 1: ANNUITY.CBL


```
VIEW          KC03C59.COBOL.CBL(ANNUITY) - 01.48          Columns 00001 00072
****** *************************** Top of Data *********************************
000001          IDENTIFICATION DIVISION.
000002          PROGRAM-ID. ANNUITY.
000003          DATA DIVISION.
000004          WORKING-STORAGE SECTION.
000005          01  TOTAL-LOAN        PIC $$$$$$$9.99 VALUE ZERO.
000006          01  LOAN-AMMT         PIC S9(9)V9 VALUE ZERO.
000007          01  INTEREST-RATE     PIC ZZ9V99 VALUE ZERO.
000008          01  INTEREST-RATPS    PIC ZZZZ9.99 VALUE ZERO.
000009          01  INTEREST-RATEP    PIC S9(5)V99 VALUE ZERO.
000010          01  MONTHS            PIC 999 VALUE ZERO.
000011          01  MAX-YEARSV        PIC S999 VALUE ZERO.
000012          01  YEARS             PIC 999 VALUE ZERO.
000013          01  MAX-YEARS         PIC ZZ9 VALUE ZERO.
000014          01  MONTHLY-PAYMENT   PIC 9(9)V99 VALUE ZERO.
000015          01  MONTHLY-PAYSTR    PIC $$$$$$$9.99 VALUE ZERO.
000016          01  VALID-INPUT       PIC 9 VALUE ZERO.
000017          PROCEDURE DIVISION.
000018          DISPLAY "WELCOME TO ADAM W.'S PANNUITY CALCULATOR!".
000019          PERFORM UNTIL VALID-INPUT = 1
000020            DISPLAY 'ENTER PRINCIPAL LOAN AMOUNT: '
000021            ACCEPT TOTAL-LOAN
000022            COMPUTE LOAN-AMMT = FUNCTION NUMVAL(TOTAL-LOAN)
000023            MOVE LOAN-AMMT TO TOTAL-LOAN
000024            IF (LOAN-AMMT < 10000000 AND LOAN-AMMT > 0)
000025              ADD 1 TO VALID-INPUT
000026            ELSE
000027              DISPLAY 'ERROR: PLEASE ENTER A POSITIVE VALUE < 10M!'
000028            END-IF
000029          END-PERFORM.
000030          ADD -1 TO VALID-INPUT.
000031          PERFORM UNTIL VALID-INPUT = 1
000032            DISPLAY 'ENTER INTEREST RATE AS A DECIMAL (1% = 0.01): '
000033            ACCEPT INTEREST-RATE
000034            COMPUTE INTEREST-RATEP = FUNCTION NUMVAL(INTEREST-RATE) * 1
000035     -        00
000036            IF (INTEREST-RATEP < 30 AND INTEREST-RATEP > 0)
Command ===>                                              Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand     F5=Rfind     F6=Rchange
 F7=Up        F8=Down      F9=Swap      F10=Left      F11=Right    F12=Cancel
       0.2 09/23/24.267 12:55PM zos.ccec.unf.edu          a     41/7
```
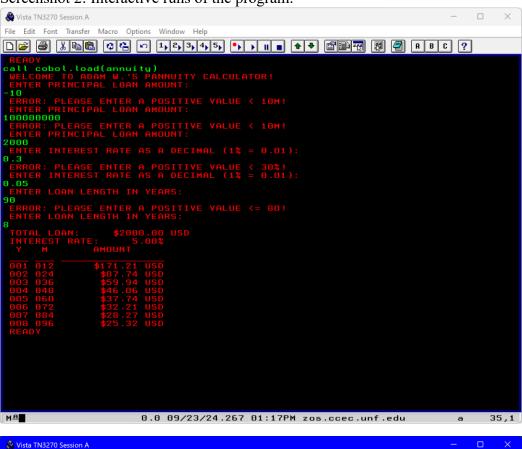


```
VIEW          KC03C59.COBOL.CBL(ANNUITY) - 01.48          Columns 00001 00072
000037            ADD 1 TO VALID-INPUT
000038            ELSE
000039              DISPLAY 'ERROR: PLEASE ENTER A POSITIVE VALUE < 30%!'
000040            END-IF
000041          END-PERFORM.
000042          ADD -1 TO VALID-INPUT.
000043          PERFORM UNTIL VALID-INPUT = 1
000044            DISPLAY 'ENTER LOAN LENGTH IN YEARS: '
000045            ACCEPT MAX-YEARS
000046            COMPUTE MAX-YEARSV = FUNCTION NUMVAL(MAX-YEARS)
000047            IF (MAX-YEARSV < 81 AND MAX-YEARSV > 0)
000048              ADD 1 TO VALID-INPUT
000049            ELSE
000050              DISPLAY 'ERROR: PLEASE ENTER A POSITIVE VALUE <= 80!'
000051            END-IF
000052          END-PERFORM.
000053          MOVE INTEREST-RATEP TO INTEREST-RATPS.
000054          DISPLAY 'TOTAL LOAN: ' TOTAL-LOAN ' USD'.
000055          DISPLAY 'INTEREST RATE: ' INTEREST-RATPS '%'.
000056          DISPLAY ' Y    M        AMOUNT        '.
000057          DISPLAY '_____'.
000058          PERFORM MAX-YEARSV TIMES
000059            ADD 12 TO MONTHS
000060            COMPUTE MONTHLY-PAYMENT = LOAN-AMMT * FUNCTION ANNUITY ((FU
000061     -        NCTION NUMVAL(INTEREST-RATE) / 12), MONTHS)
000062            COMPUTE YEARS = MONTHS / 12
000063            MOVE MONTHLY-PAYMENT TO MONTHLY-PAYSTR
000064            DISPLAY YEARS ' ' MONTHS ' ' MONTHLY-PAYSTR ' USD'
000065          END-PERFORM.
000066          GOBACK.
****** *************************** Bottom of Data *****************************




Command ===>                                              Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand     F5=Rfind     F6=Rchange
 F7=Up        F8=Down      F9=Swap      F10=Left      F11=Right    F12=Cancel
       0.0 09/23/24.267 01:03PM zos.ccec.unf.edu          a     41/7
```

Screenshot 2: Interactive runs of the program.

Adam Walton

Project 2

CIS 4930 COBOL for FinTech

Comments:

I found this project to be more challenging than previous assignments. I had to ask for help with my JCL, since I still found it confusing, and found out that the program ID in the COBOL code didn't match the program name in the JCL file. I also had trouble dealing with input and output, since COBOL treats numeric values much differently than other newer languages do. Upon finding this out, I looked up the issue and learned about a "NUMVAL" function that could convert decimal strings to numeric data, as well as add leading/trailing zeros if the user input "2000" instead of making them enter something like "000002000000" to represent a $200 principal amount and prevented any abends that occurred from decimal value input.

However, I couldn't find a way to check if a decimal value was numeric, and later learned that was a tradeoff in where I couldn't check if data is numeric data if the user was allowed to enter values without dealing with leading zeros, and vice versa. I was told there was a way to do this, but I couldn't find anything online to fix the issue. Since I wanted to prioritize clean looking user input over checking for numeric data, I decided to implement checks to see if users entered in data over a certain limit, or if they entered a negative value and loop until they enter valid data, which worked out well.

Overall, I am happy with the outcome of this project. I think it helped me learn much better how COBOL represents different types of data, COBOL's loops and conditionals, and how JCL might work. I also learned that logging off the mainframe saves search filters, which caused me to not see my compile jobs until I figured out how to clear them.