Phase 3 By Team Alpaca.

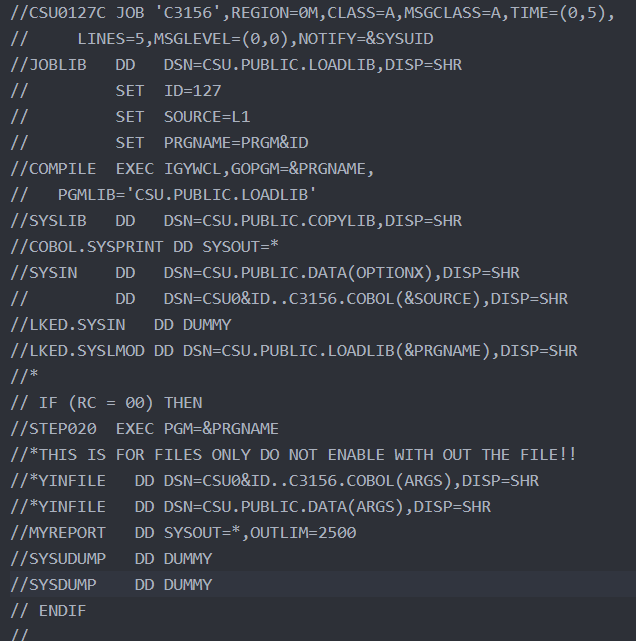
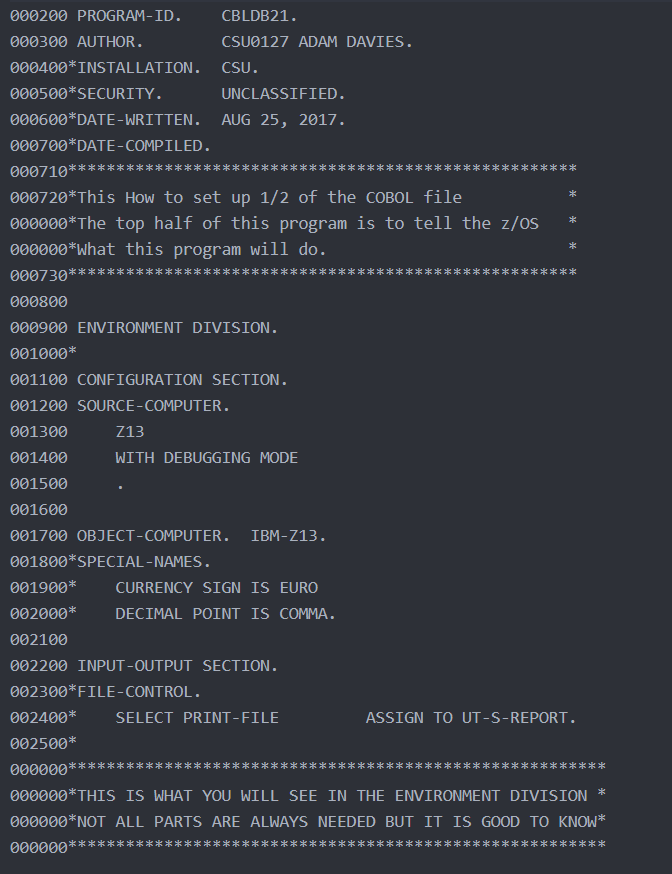
Do to the nature of COBOL requiring a lot before code before you can start coding we set up the assignments where they build on top of each other. The first Assignment is JCL which we are aware of that it is not COBOL but it must be written for any COBOL program to work thus we put it in. this program is called R1.jcl in the GitHub repo and it will run the COBOL programs that will be written later on. A sample of R1.jcl is Figure 1.

Figure This is R1.jcl



L1 is the first COBOL program that was written this is mostly for the user to copy and past just like the JCL file and it has a lot of terms that is required for the program to run. A sample of L1.cbl is on the right.

L2 is the second COBOL program to be written this program is where the we add in some variables into the data division under the storage section and build the procedure division for Hello world. Hello world works by moving the string “Hello World” into a variable. Later, we use the key word display to show the variable once the program is run. In Figure 3 you can see the out put of running the program.

Figure This is some of L1.cbl

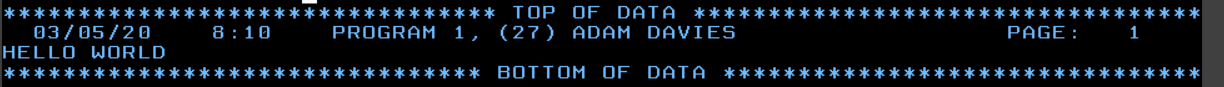
L3 is the third and last COBOL program that is easy. This is to place two strings together like “Hello world” and “Team Alpaca”. 

Figure L2 output after running it

M1 is the first mid-level COBOL program. In this program we will display more then one line out to the end user. This is just performing the action more then once.

M2 is the COBOL program where we introduce numbers and perform addition and subtraction. The coder will need to add number fields in the data section using the pic 9(14). Will work for this after that the coder can right the addition to the fields by ADD 1 TO VAR1 note this will put the sum into VAR1.

M3 is the COBOL program where we continue on numbers by performing multiplication and division. This time instead of righting over the VAR1 with the math functions we can MULTIPLY 40 TO VAR1 GIVINING VAR2. This will put the new number in to VAR2 instead of writing over VAR1 “GIVINING” is the key word for doing this.

H1 is the start of the Hard COBOL programs. In this program we will use the if command. The end-if is needed to show that you have ended your if condition a period will work but it is somewhat bad practice to do that also you may find yourself where you want to right more to that paragraph and using a period will prevent you from being able to do that.

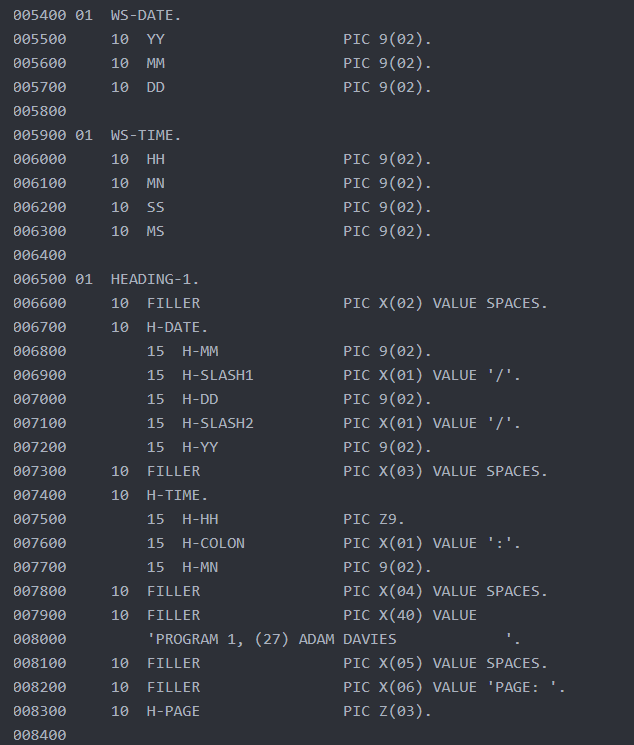
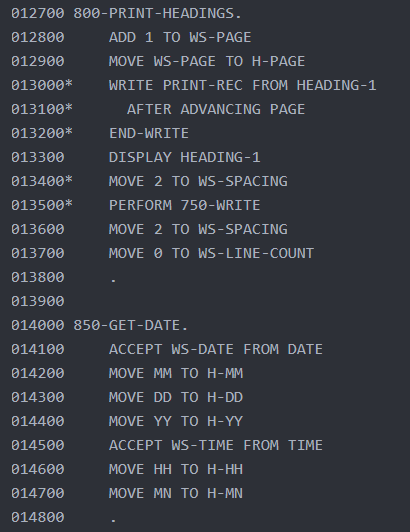
H2 is the program that will make a header that will show the current date by getting it from the system. There is samples of how this is done with Figure 3 and Figure 4.

Figure H2.cbl This is the procedure div side

Figure H2.cbl on what is needed on the data side

H3 is the program will use files instead of the display function. The file that are being used must be made before running the COBOL program and the data must be populated first. Note it is important on this that the JCL is edited the place where the “ARGS” are at needs to be the file output name with out that the program will have a runtime error. I think 08 error if I remember right.