

Aaron Valenzuela
Sean Williams
Michael Glenn

COBOL Data Types

Cobol has four data types available for the programmer to use: floats, ints, strings, and arrays. Arrays can hold items of the three other data types. The main difference between ints and floats in COBOL is that floats can be accurate to a decimal point of one's liking as defined in the assignment with a 'V' followed by the number of integers being the level of precision. In COBOL, all variables that will be used in the program must be declared in the DATA DIVISION WORKING-STORAGE SECTION. Below is an example of all four data types being initialized.

```
DATA
DIVISION.

    WORKING-STORAGE SECTION.
01 ValueInt PIC 999.
01 ValueFloat PIC 9(3)V99.
01 ValueString PIC X(20) VALUE "Hello".
01 ARRAY-STRINGS.

    05 ARRAY-ITEM PIC X(3) VALUE "Hi!" OCCURS 5 TIMES INDEXED
    BY I.
```

Integer and float data types are declared by reserving an amount of space for the amount of integers to be used for the data. In the first line, three spots are saved by writing PIC 999. This can also be done by 9(3) for the sake of simplicity when reading the code this will result in the value always being at least three characters in length even if the value is 1000 or 0 the result would in both cases be 000. The first variable, ValueInt, is an integer of up to three digits pre decimal and two digits post decimal. So, the largest value that can be stored in ValueInt is 999.

Increasing the amount of 9s or the number in the parentheses increases the integer size of the variable. The same concept works for the float types except for the addition of the V character. In the second line, the number of 9s after V represents the number of decimal places to the right that is desired for this float variable. So ValueFloat will be calculated to the second decimal place (one hundredth place) reaching a max value of 999.99. There are two ways to perform arithmetic operations on integer and float data types:

```
ADD 35 TO ValueX GIVING ValueX.  
COMPUTE ValueX = ValueX + 35
```

The two operations above both add 35 to a variable named ValueX. Dividing, multiplying and subtracting are done the same way for the COMPUTE function, just enter whatever operation you want done using the keyboard. ADD 35 TO ValueX is the operation that is being performed, and whatever variable is after the GIVING will save the result of that operation. There are also MULTIPLY, DIVIDE, and SUBTRACT functions following the same format, just replace ADD with the desired function and the order of operations are performed just as they are read.

The string data type above named ValueString contains the value “Hello”. Strings in COBOL can be concatenated. An example is shown below:

```
STRING ValueString DELIMITED BY SPACE "HELLO WORLD!" DELIMITED BY SIZE INTO  
ValueString.
```

This is equivalent to doing: ValueString += “Hello World!”

So, the result of this statement would be: “HelloHello World!”

Arrays in COBOL have items placed in them when being initialized. In our array of strings above, “Hi!” is initialized as five times in the array by the line: VALUE “Hi!” OCCURS 5 TIMES INDEXED BY I. The ‘I’ character will be used as an index for the array. Arrays can be indexed using ARRAY-NAME(index_int). When indexing an array it is important to know that indices start at 1 and not 0 unlike other programming languages. The line below changes the first value in our string array to the string value “NEW”.

```
MOVE "NEW" TO ARRAY-ITEM(1)
```

To iterate through an array using a loop, use the variable name after the statement INDEXED BY when the array was first initialized to correctly start from the beginning of the array. An example of this is shown below.

```
PERFORM OUR-DISP VARYING I FROM 1 BY 1 UNTIL I>5
```