BCI433 Lab 9 (updated Summer 2023)

**Converting Lab 3b to employ user defined functions**

**Lab objective:**

* **Copy, study put together and run a user defined function sample**
* **Create user defined functions for the current Lab 3 Application**

**Lab Requirements:**

**Show what UDF’s you constructed for the Lab 3 App**

**show where the UDF is used and show the UDF solution code**

**Demonstrate how to put everything together**

**Successfully run Lab 3 App that now employs user defined functions**

User Defined Functions

A user defined function example is available for you to copy, study, compile and run

.

Text

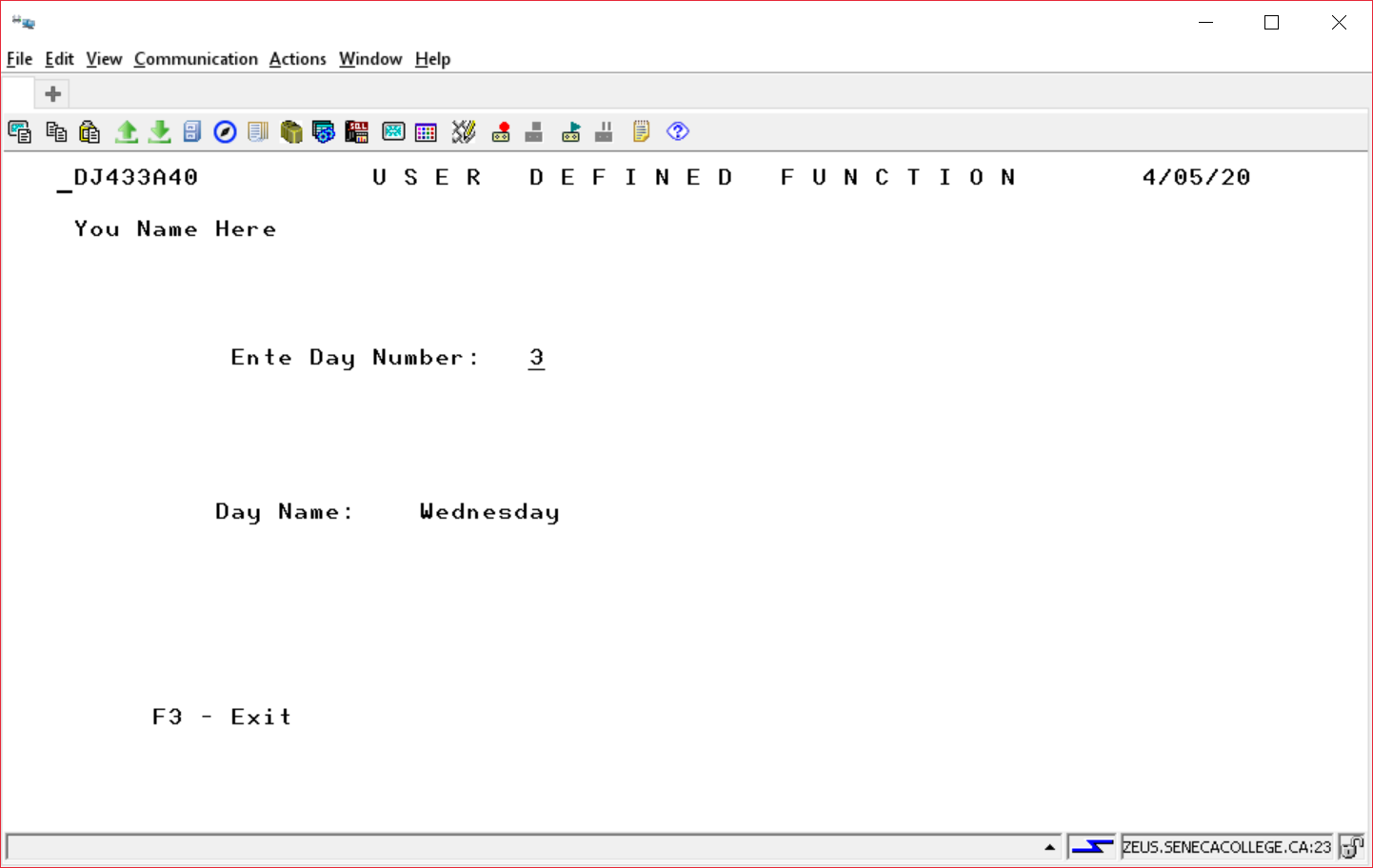
Description automatically generated with medium confidence

DayNumName( ) accepts a one digit number and returns a day name word. First this user defined function is set up using inline case structure as a simple solution and that code is entered in DAYFUNCTS.

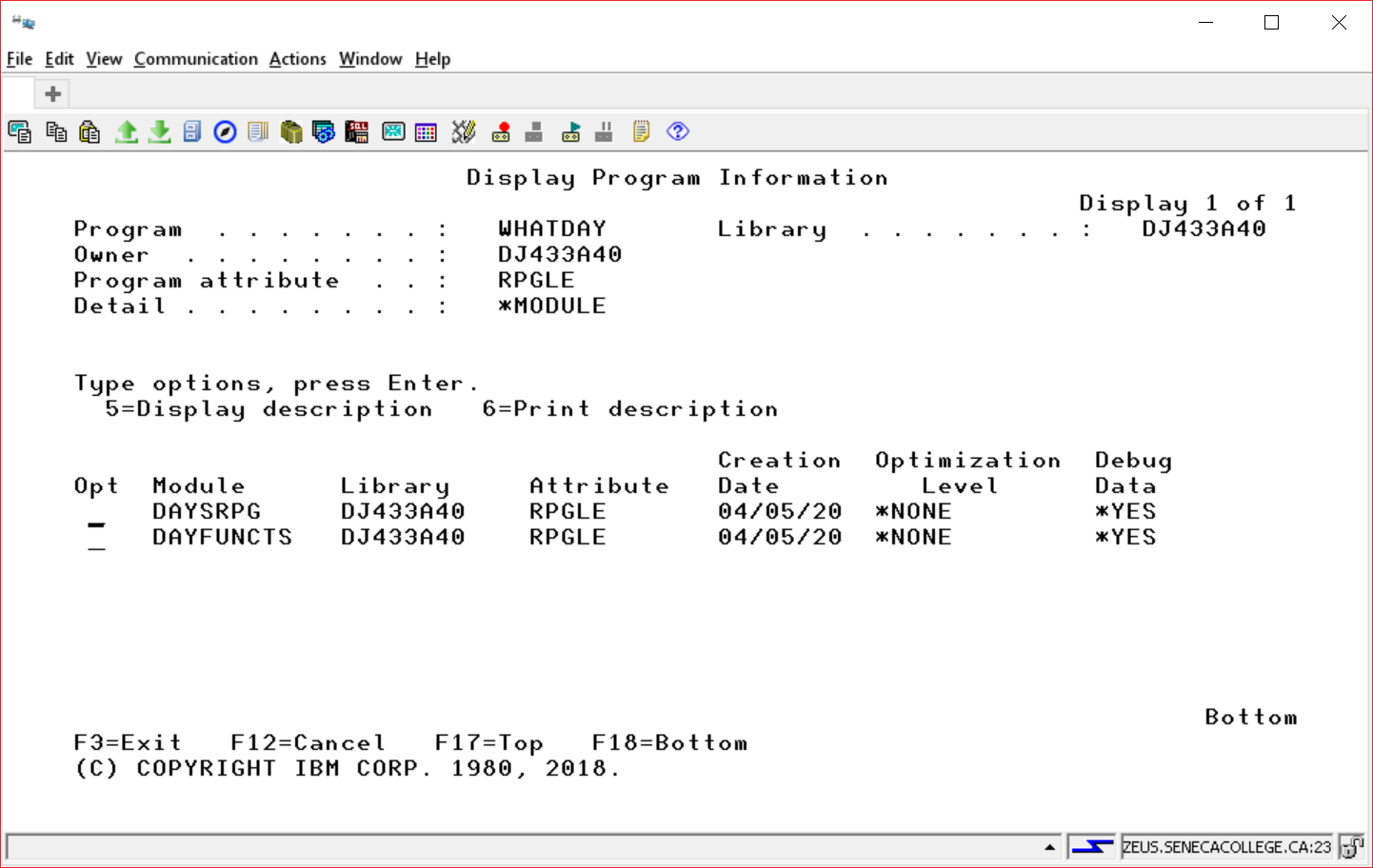
A better solution using an array is determined after the Case structure is used for a period of time. That code is entered in DAYFUNCTS2.

When you copy this code, you should try compiling the modules and then binding them together without the help of the CLLE program so you know how to deal with your own user defined functions. The whole process is automated in a CLLE program called RUNUDFDEMO which also provides some limited information as it runs. The scenario is to produce the first working program using the simple case solution code and then see that replaced with code that uses a more efficient arrray solution. The case solution insulates the user from an unexpected entry like a day number of 8 when there are only 7 day names in the week. The array solution purposely does not compensate for an entry of 8 and will bomb if the day number entered is not a digit from 1 to 7.

Input and Output records for DAYSRPG which relies on a user defined function



The module that relies on a user defined function is combined with the module that solves the user defined function into a working program called WHATDAY



All the code is available to be copied, but lets focus on the user defined function code.

DAYSRPG

DCL-f DayDsp Workstn;

// copy the prototype into your program that uses the user defined function

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EXFMT INPUT;

DOW NOT(\*IN03);

// the user defined function

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*in99 = \*on;

WRITE INPUT;

EXFMT OUTPUT;

\*in99 = \*off;

IF \*IN03 = '0';

DayIn = 0;

EXFMT INPUT ;

ENDIF;

ENDDO;

\*INLR = \*ON;

RETURN;

DAYPROTO contains two prototypes

The Prototype for DayNumName

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The Prototype for MonthNumName (not used in this lab)

Used to support a more sophisticated Day Name like

Sunday April 5, 2020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DayNumName and MonthNumName are not RPGLE functions. But, they can be created and then used by all programmers. The RPGLE compiler would reject

DayName = DayNumName(DayIn);

The prototype is telling the compiler to accept DayNumName as legitimate.

DAYFUNCTS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (no need for \*INLR = \*ON)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Get the Prototype code)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( the user defined function)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (what is returned from the function)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (what is input to the function)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Local variable)

(Solve the function)

SELECT;

WHEN NUMBER = 1;

DAYNAME = 'Monday';

WHEN NUMBER = 2;

DAYNAME = 'Tuesday';

WHEN NUMBER = 3;

DAYNAME = 'Wednesday';

WHEN NUMBER = 4;

DAYNAME = 'Thursday';

WHEN NUMBER = 5;

DAYNAME = 'FRIDAY';

WHEN NUMBER = 6;

DAYNAME = 'Saturday';

WHEN NUMBER = 7;

DAYNAME = 'Sunday';

OTHER;

DAYNAME = 'Unknown';

ENDSL;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DAYFUNCTS2

This program will be discussed in class. It will replace the in line case structure with a positional array lookup. It will not error check for an invalid number like the solution above did. It can be set up so it will be impossible to send the day function a weekday number that did not exist (like 9) from the program using this function solution.

Dcl-DS DayData;

\*n Char(9) Inz('Monday');

\*n Char(9) Inz('Tuesday');

\*n Char(9) Inz('Wednesday');

\*n Char(9) Inz('Thursday');

\*n Char(9) Inz('Friday');

\*n Char(9) Inz('Saturday');

\*n Char(9) Inz('Sunday');

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // Define the arrray

End-DS;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // Use the array

All the code demonstrating a user defined function to convert a day number into a word is available to be copied and run as shown at the top of this lab.

Here is an example of a program used in a previous semester for lab 3 which has been converted to employ user defined functions.

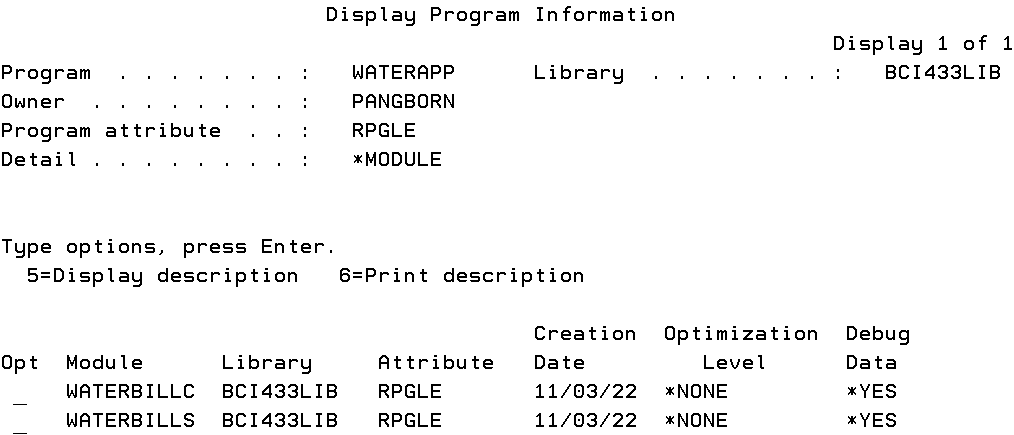
You can run this program:

CHGCURLIB BCI433LIB

CALL WATERAPP

You can investigate the modules in WATERAPP

DSPPGM BCI433LIB/WATERAPP DETAIL(\*MODULE)



WATERBILLC is the module that employs two user defined functions





WATERBILLS is the module that contains the procedures that solve for the UDF’s

Bothh WATERBILLC and WATERBILLS copy in the prototypes.



Your task is to determine eligible code from your LAB3B that can be solved with a user defined function.