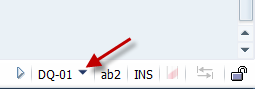
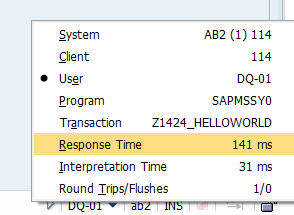
**Note the following instruction to be followed throughout the semester:**

When submitting screen shots of your lab work, be sure that your user name is visible in the lower right of the screen capture. To accomplish this:

(1) If the lower right screen region is not displaying status text, click the left-pointing caret in that region to open the status information,



(2) If your user name is not a part of the display, click the downward-pointing caret and select “User”.

**(3) Your program names are at your discretion, however each should begin with ##### (where #### is replaced by your login name without the -).** This will help avoid name collision within the system. Points will be deducted from program submissions where this naming convention is not followed. Follow this naming convention for the duration of the semester, realizing that future homework instruction sheets will not remind you of this requirement.

For all homework this term:

Once you have finished developing your program, **activate the final version** so that it can be run tested for grading. You do **not** need to install the program as a transaction code, nor should you transport your program.

To submit your work, upload to D2L a Microsoft Word document containing the following:

* The name of your program. The program may be run-tested for grading.
* A copy of your ABAP source code. Copy the code in its entirety (omitting the large comment block at the top) and paste it into the Word document **as text**.) If your code lines of code are so long that Word breaks them in odd places, change the format of the document to landscape.
* A screen shot of your application’s **output** shown running in the SAP GUI. Use as many screen shots as necessary to capture all output. Be sure to show the entire window—do not crop the image.
* Limit the materials included to just the above or other specific items noted in the instructions. Do not include extra materials or screen shots of things other than the above.

Program One:Design a structure, fill it with data, and display output

**It is recommended that you review the requirements of this program several times during your work and then also review your final submission candidate to insure that you have fulfilled everything to the best of your ability.**

A library tracks the following information for all books:

* Book Title
* Primary Author
* Publisher
* Publisher State (2 letter abbreviation)
* Publication Date
* Tracking ID (this is a 9-digit numeric code such as 987654321)
* Purchase Cost
* Date of Purchase

Create a structure type capable of holding the information noted above. Take some time to consider the appropriate type choices for each field.

Create two structures and fill each of them with data for a hypothetical book. Output each of the structures in the following manner:

Label format:

Book Title: Title  
Author: Author  
Publisher: Publisher, ST  
Published: MM/DD/YYYY  
Tracking ID: xxxxxxxxx  
Purchase Cost: $x.xx  
Date of Purchase: MM/DD/YYYY

Table format:

Title Author Publisher State Published Tracking ID Cost Purchased

Title Author Publisher NY MM/DD/YY xxxxxxxx $x.xx MM/DD/YY  
Title Author Publisher TN MM/DD/YY xxxxxxxx $x.xx MM/DD/YY

Note the following:

* Write both of the books out in label format with a blank line between then. Then follow that output with a single table that contains information on both books. (One table header with two books in table body.)
* Match the formatting shown here precisely. The horizontal spacing of the table is at your discretion, but make it look nice. Pay particular attention to the dates.
* Do not store anything in your structure that does not need to be there. Certain elements of the above are added in the output statements as literal text.

Program Two: As demonstrated in class, a number of different system-maintained data values are maintained by the ABAP environment within the SYST structure. Use that structure to display the attributes of the system you are using this semester. (Use the Data Dictionary [SE11] to display data type SYST to determine the appropriate fields to use within the structure.)

Place your output into 3 columns. The first column will contain the name of the data object in the system, the second column will contain the value of the data object, and the third column will contain a brief description of the data object. Make your output look neat by aligning all of your columns. Place a header line—as shown below—at the top of each column to describe the output.

*Note that while it is possible to construct this output in a variety of ways, the output statements that are used to build the rows that contain the field-name, field-value, and text description (i.e. the second through nth rows of the table) should* ***not*** *employ position or length specifications.*

For example: Your output should resemble the following exactly. (Horizontal space between columns is at your discretion.)

Field Name Value Description

SY-XXXX Mario Name of protagonist in Donkey Kong  
Field-name Field-value Text description of field  
Field-name Field-value Text description of field  
  
You will notice that there are 171 different fields in SY. Many are obsolete and many are highly specialized for purposes we will see later this semester. However, there are fields that allow us to find out the login name of the current user, the client number, and the database system in use.

Look through SY and find **6 data values** that have some potentially useful information in them to be displayed. Make these selections the output of your table.

Note the following:

* The text displayed in column 1 and column 3 is literal text, supplied by you at coding time. There is no need for this text to be stored in data objects.
* The body of your table should be output using the chained statement technique where each line of output corresponds to one WRITE statement.

Program Three:Write a program which will receive from the user at run time two integer numbers and an operator (+, -, \*, or /). Display the result of using the operator on the two numbers. Display both the equation and the answer.

Example output if the user enters the numbers 3, 7, and the + operator:

3 + 7 = 10

Use a CASE statement to determine the calculation for the entered operator. Be sure that the results displayed are correct. 5 / 2 should not display 2 as the answer.

Various elements of your design are not explicitly enumerated here but should come from your complete reflection on the task to be completed. Both your code itself and the overall design of your solution are elements in the grade.

**Submit your work to D2L following the instructions provided in page 1 of this document. (All programs should be contained in one Word document.) Submit to the Lab 2 dropbox in D2L by the due date posted in D2L.**