CASE 4: DIALOG PROGRAMMING

Requirement

Create an ABAP program that will let you perform simple dialog programming.

Process

1. Create a dialog Program
2. Create PF-STATUS for the ADDITION, SUBSTRACTION, MULTIPLICATION, DIVISION
3. Create input field for the First and Second Parameter
4. Create button for execute computation and clear results

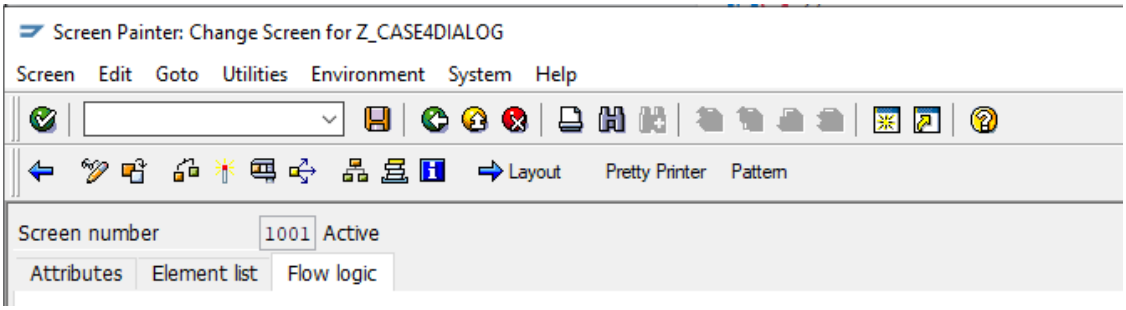
Notes: the input parameter should only contain numeric

First, we are creating Screen for our dialog

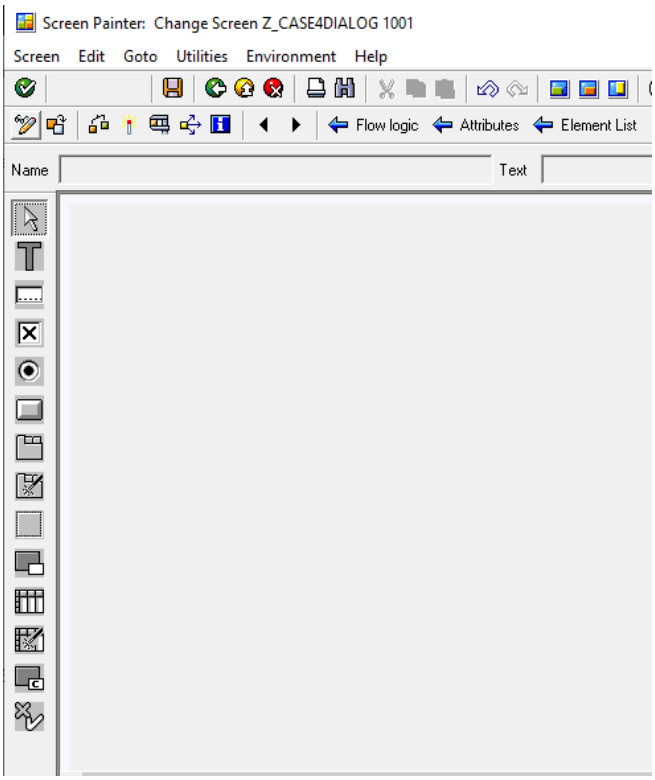
```
CALL SCREEN 1001.
```

Then, double click the variable name ‘1001’ for the screen painter to open

On the screen painter, on the toolbar, click the ‘-> Layout’

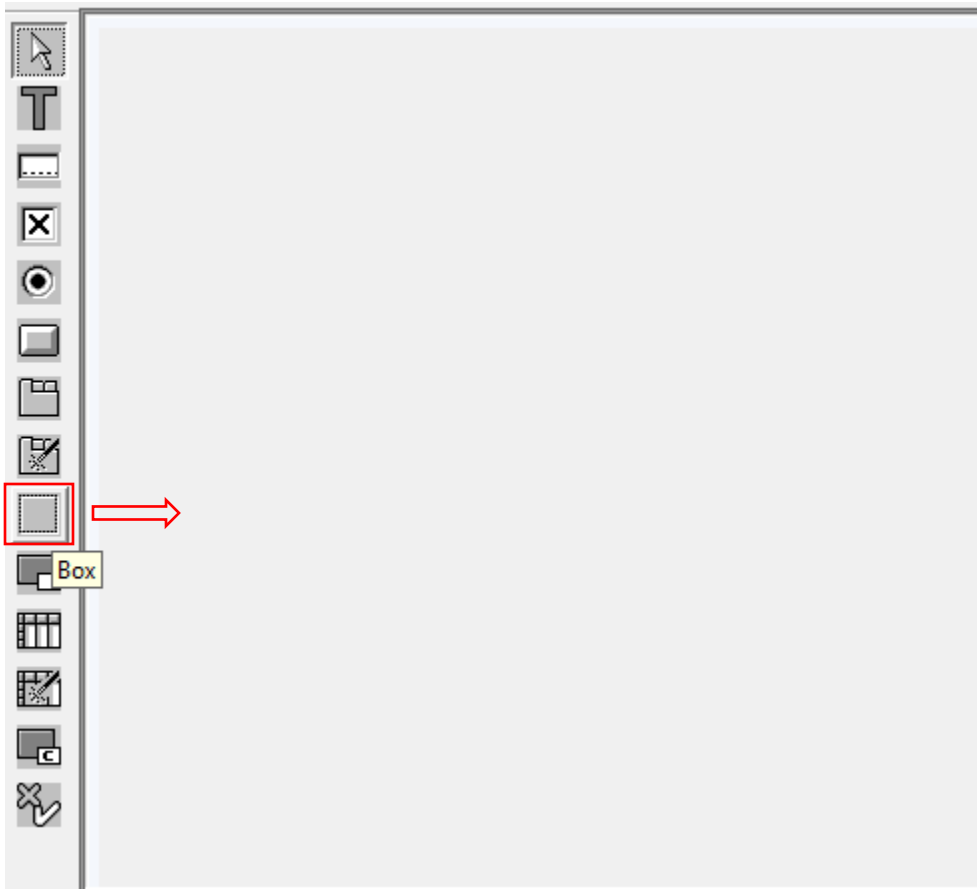


The screen painter screen will show



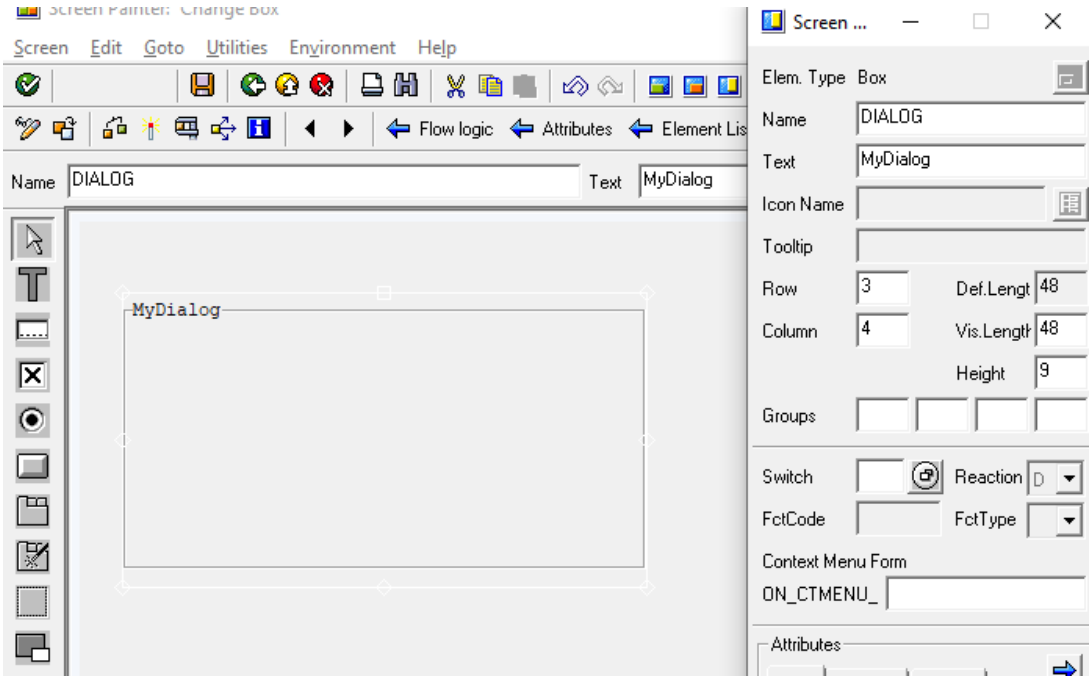
We will create our GUI

First drag a box from the icons on the right

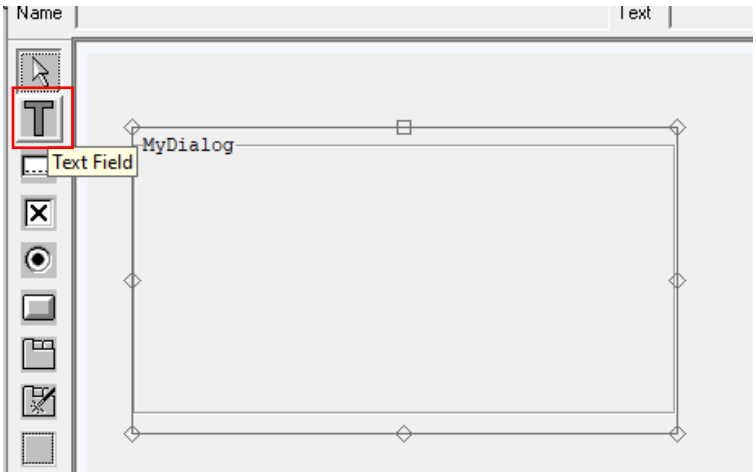


Double Click on the Box, and the attributes will show

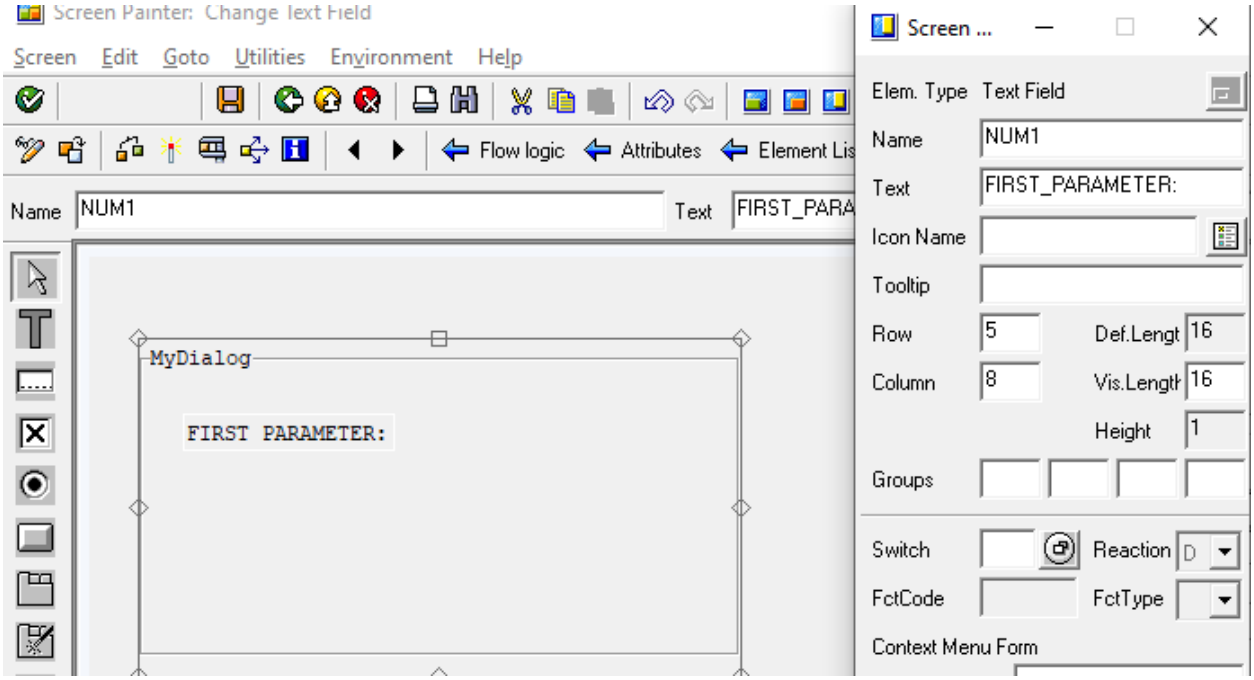
Change it to appropriate text label and name.



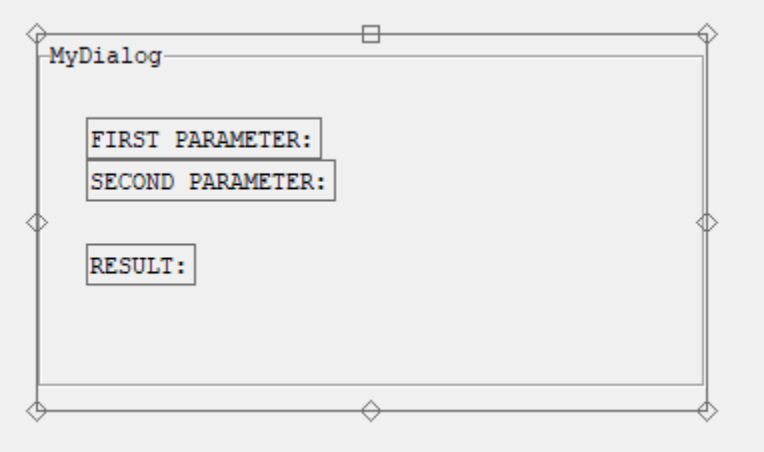
Next is to draw text fields.



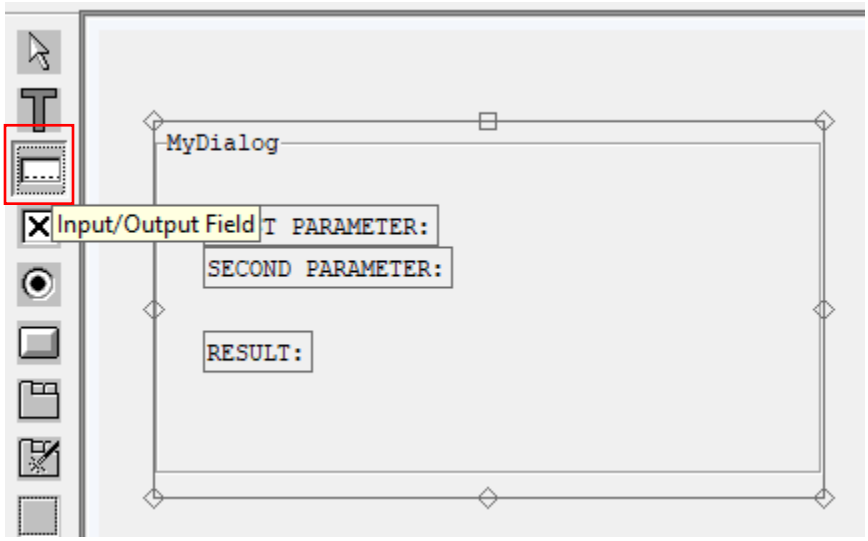
Put the appropriate name and text label:



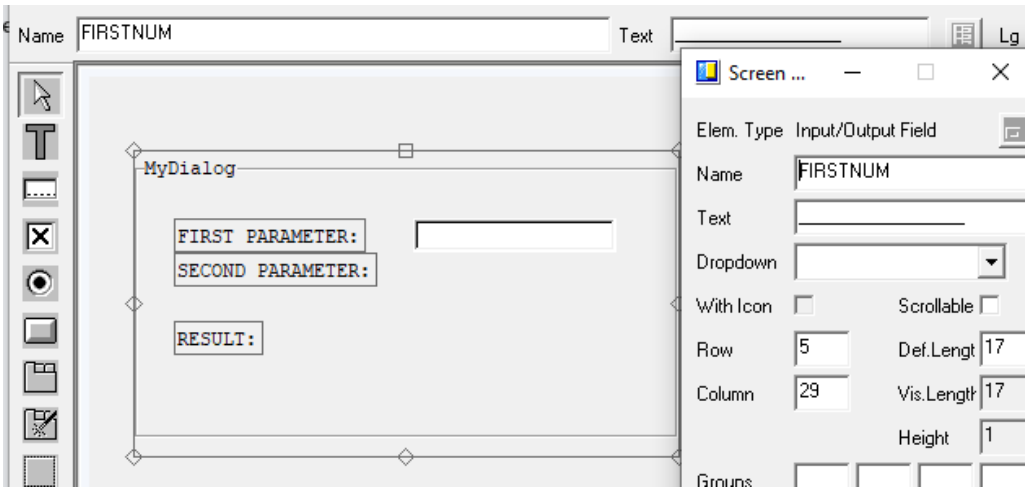
Do the same to other text fields.



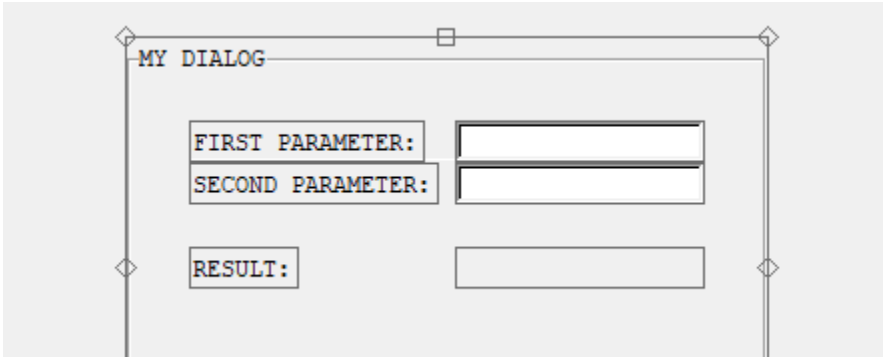
Next is to draw Input/Output field for the parameters:



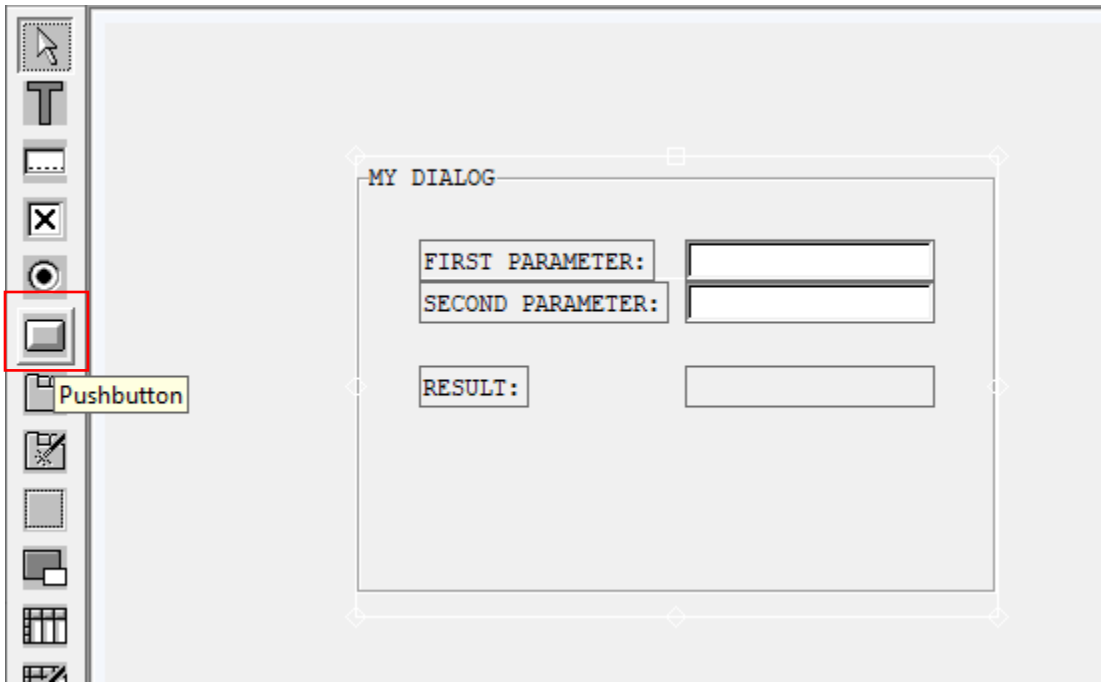
Put the attribute for the input/output field



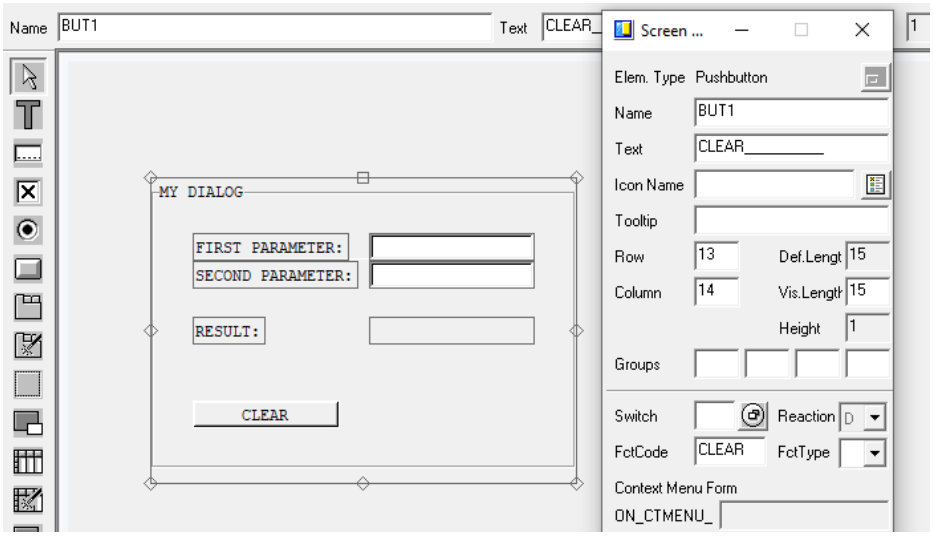
Do the same for the others.



Draw the pushbutton



Put the attributes:



To save click the activate icon on the toolbar:

Go back to the Screen Painter Main Screen, and write the PBO, and PAI for the Dialog

First is the PBO

```
1    PROCESS BEFORE OUTPUT.
2     MODULE STATUS_1001.
3
```

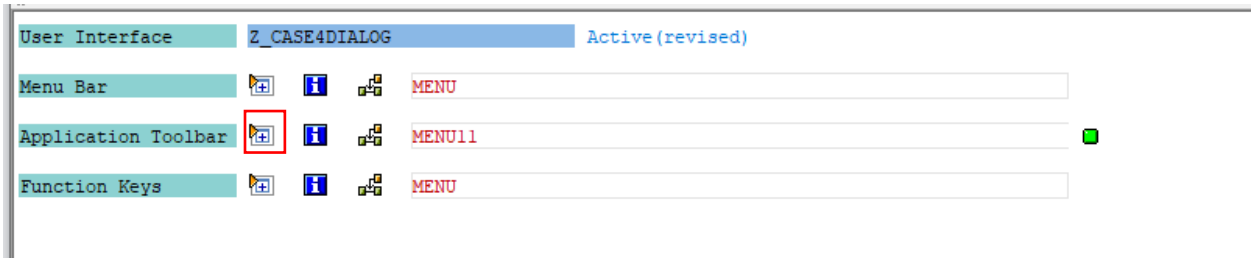
Double click on the STATUS_1001 to view the module

Then we declare the PF-STATUS

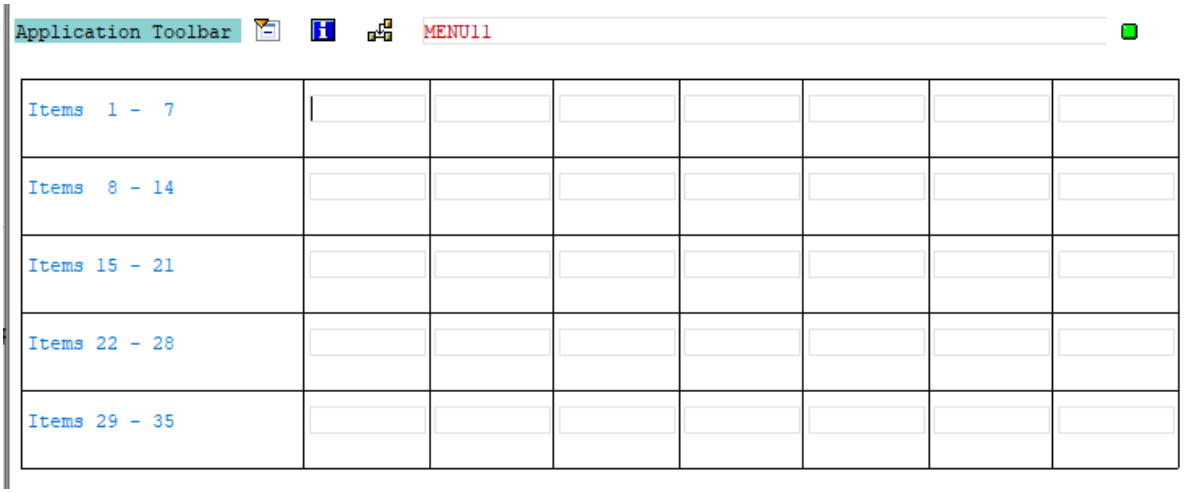
```
17    SET PF-STATUS 'Z_MENU11'.
18     SET TITLEBAR 'TITLE'.
```

Double click on Z_MENU11 the pf-status we declare to go to edit the Status of Z_MENU11

Click on the Application Toolbar [+]

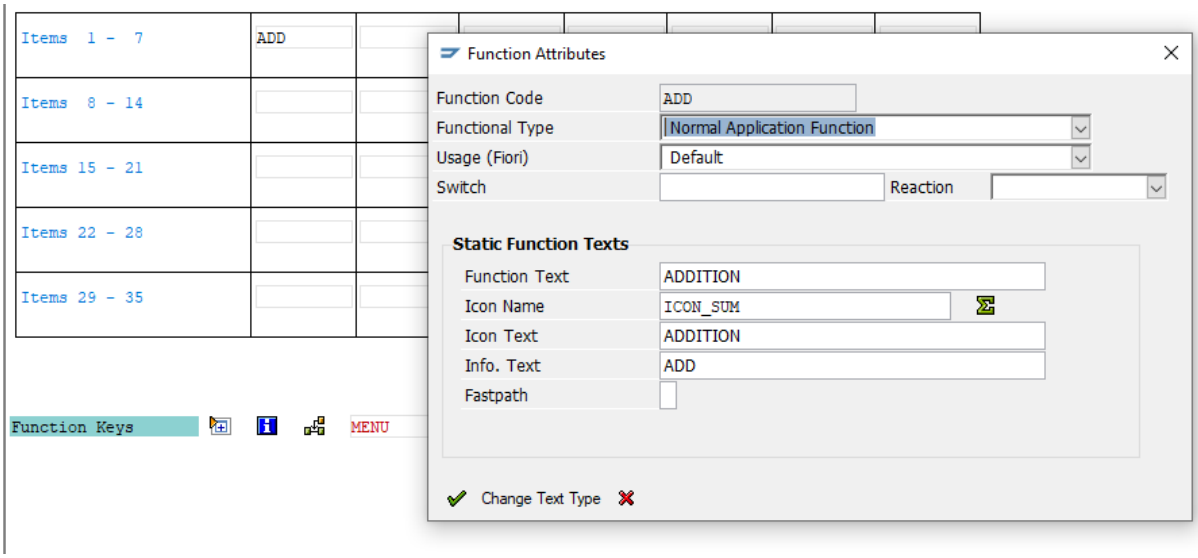


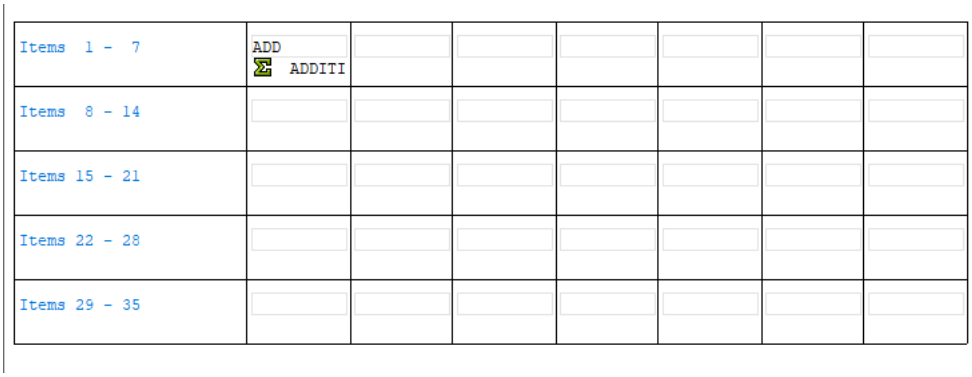
Click on the empty field to populate the item, and add the PF-STATUS ADD, SUBTRACT, MULTIPLY AND DIVIDE



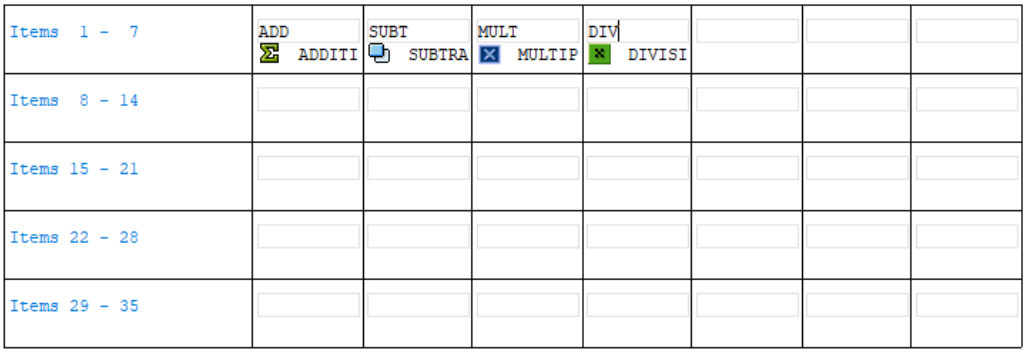
After writing the title, click below the text and the function attributes window will appear

Put the appropriate attributes

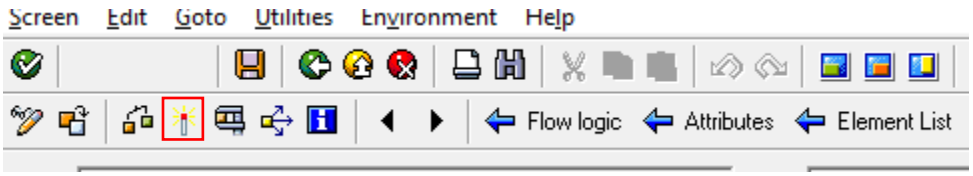




Do the same for other STATUS



To save click the activate icon on the toolbar:



Back on the PBO Module, we will write the function of the PF-STATUS

```
CASE sy-ucomm.
    WHEN 'ADD'.
      RESULT = NUM1 + NUM2.
    WHEN 'SUBT'.
      RESULT = NUM1 - NUM2.
    WHEN 'MULT'.
      RESULT = NUM1 * NUM2.
    WHEN 'DIV'.
      RESULT = NUM1 / NUM2.
    WHEN 'CLEAR'.
      CLEAR: NUM1,NUM2,RESULT.

    WHEN 'BACK'.
       LEAVE TO SCREEN 0.
  ENDCASE.

ENDMODULE.
```

Next is the PAI Module

```
PROCESS AFTER INPUT.
 MODULE USER_COMMAND_1001.
```

Double click on USER_COMMAND_1001 to view the module

Write the ‘BACK’ function so we can go back when we are testing the application

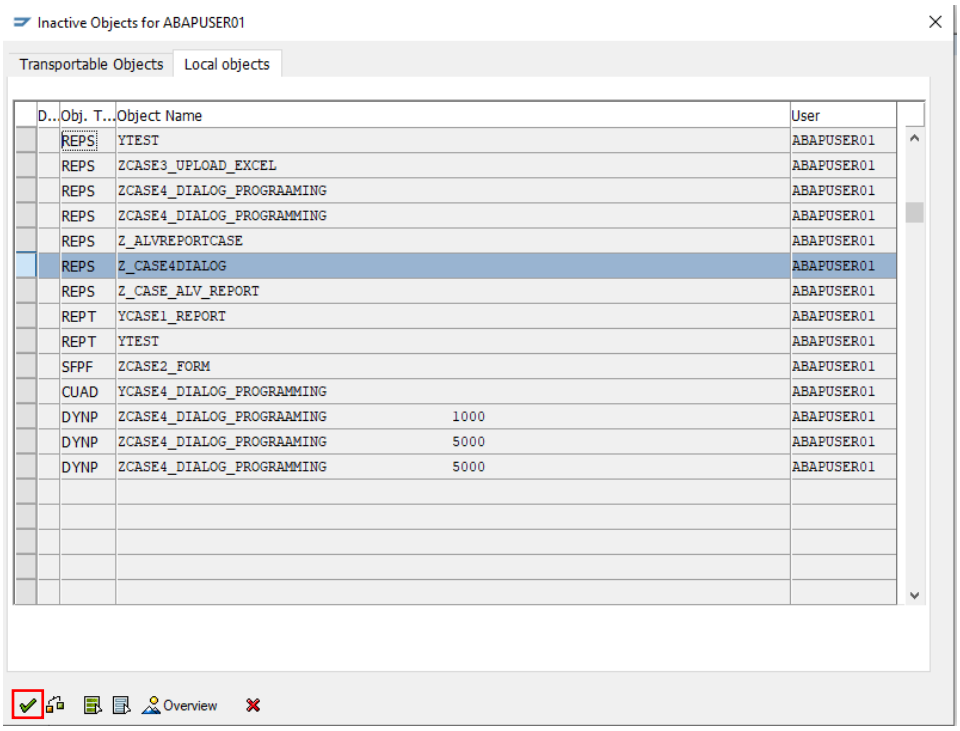
```
MODULE user_command_1001 INPUT.
  CASE sy-ucomm.
    WHEN 'BACK'.
      LEAVE TO SCREEN 0.
  ENDCASE.
ENDMODULE.
```

Now, going back to our main program we will declare the data for our parameters, remember to use the name we put on the attributes on our screen painter

```
4    DATA: num1 TYPE p DECIMALS 2.
5    DATA: num2 TYPE p DECIMALS 2.
6    DATA: result TYPE p DECIMALS 2.
7
```

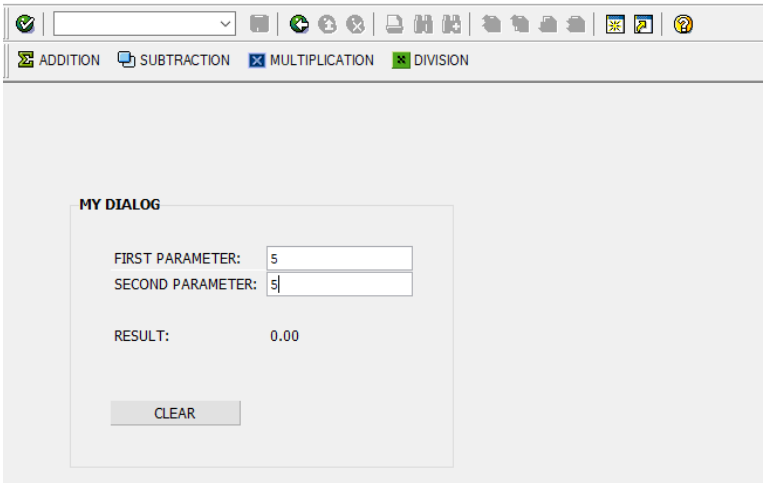
To save click the activate icon on the toolbar:

Hit Check Icon

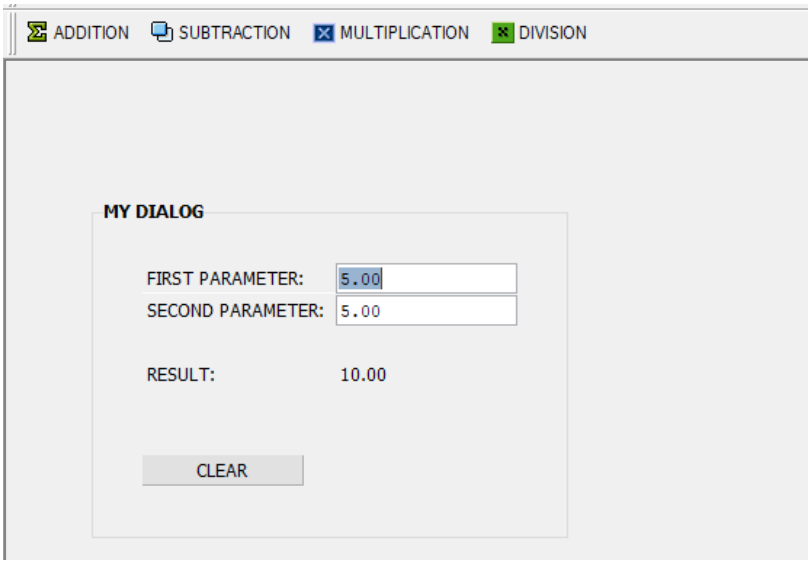


To execute click on execute icon on the toolbar:

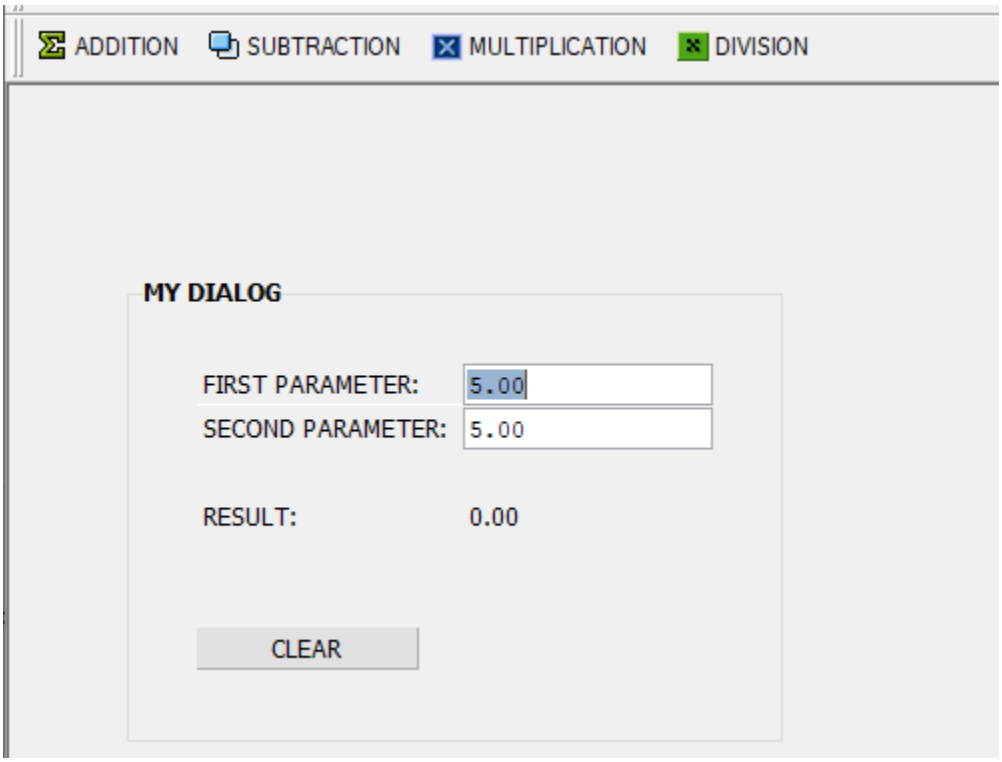
Output:



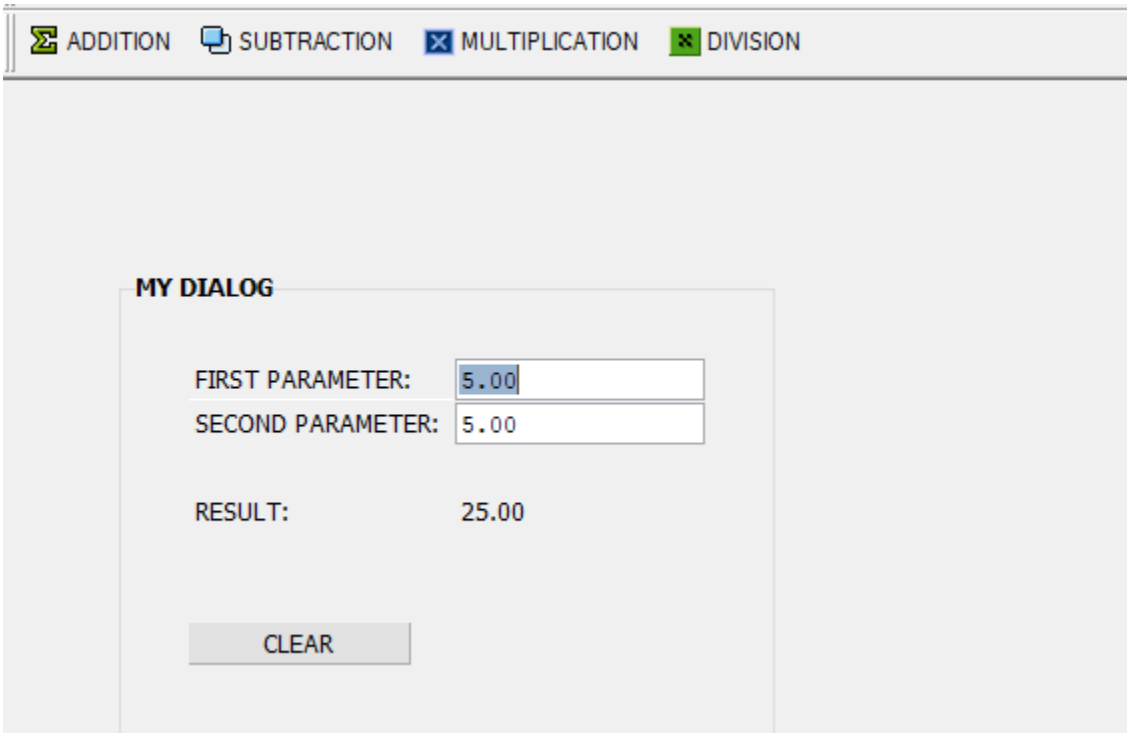
ADDITION:



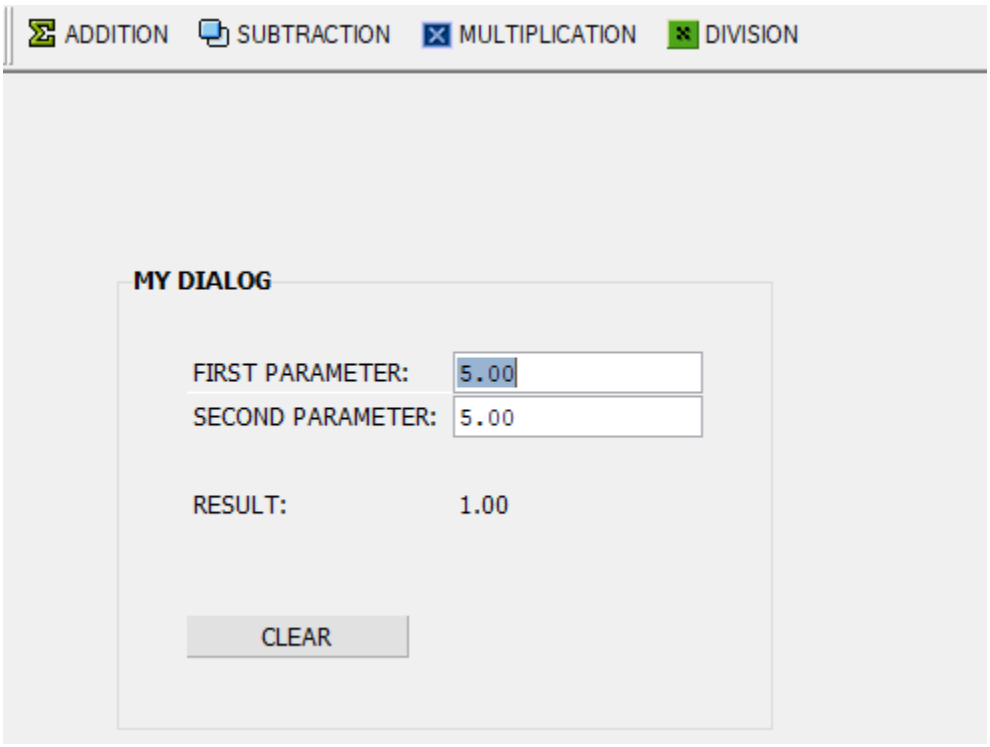
SUBTRACTION:



MULTIPLICATION:



DIVISION:



CLEAR: