CE 300 SUMMER PRACTISE
INTRODUCTION TO MATLAB

Graphical User Interface

Alper Aldemir
K2- 105

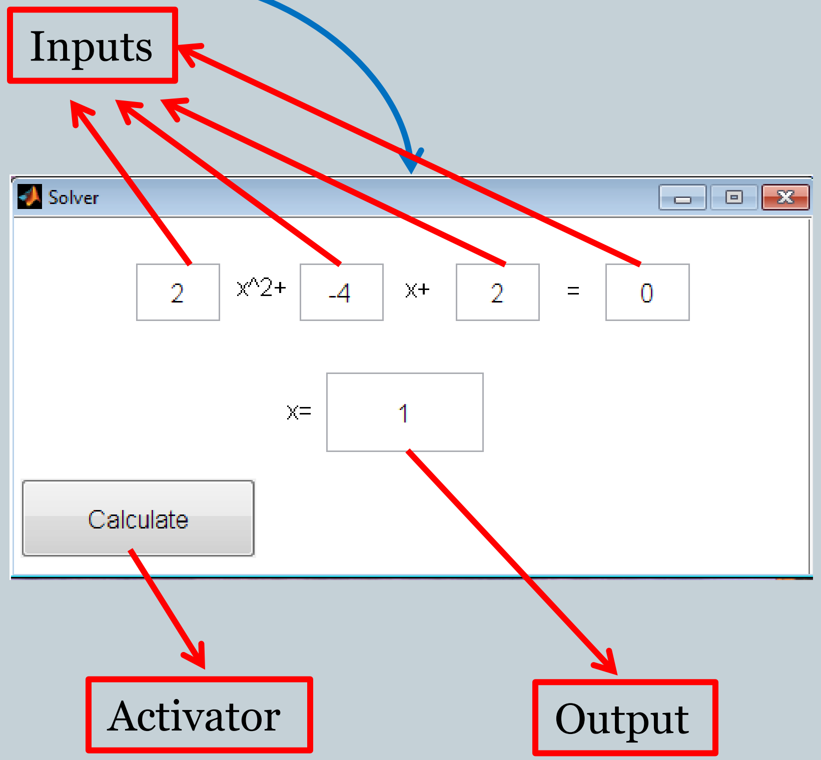
Graphical User Interface (GUI)

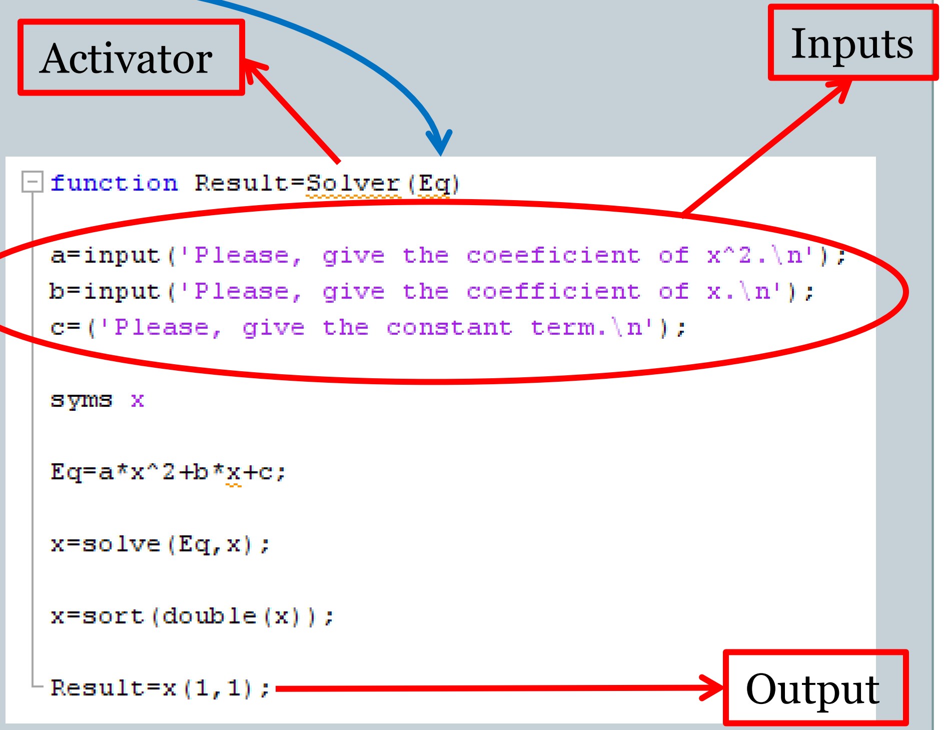
This lecture will deal with the following topics:

- Definition of GUI
- Comparison of GUI with usual scripts
- Stages in GUI
- Example (Calculator)

Definition of GUI

As a definition, **GUI** is a type of communication with electronic devices, i.e. computers, by means of **<u>images</u>** rather than **<u>text commands.</u>**





```
function Result=Solver(Eq)

a=input('Please, give the coeefficient of x^2.\n');
b=input('Please, give the coefficient of x.\n');
c=('Please, give the constant term.\n');

syms x

Eq=a*x^2+b*x+c;

x=solve(Eq,x);

x=sort(double(x));

Result=x(1,1);
```

Comparison of GUI with the usual Scripts

- GUI is very effective when the main program has to take lots of inputs from user because of its visual capability.
- It makes a complex program much more user-friendly by utilizing buttons, text boxes, check boxes, etc.
- The simple scripts can be turned into a much more better-looking ones.

Stages in GUI Design

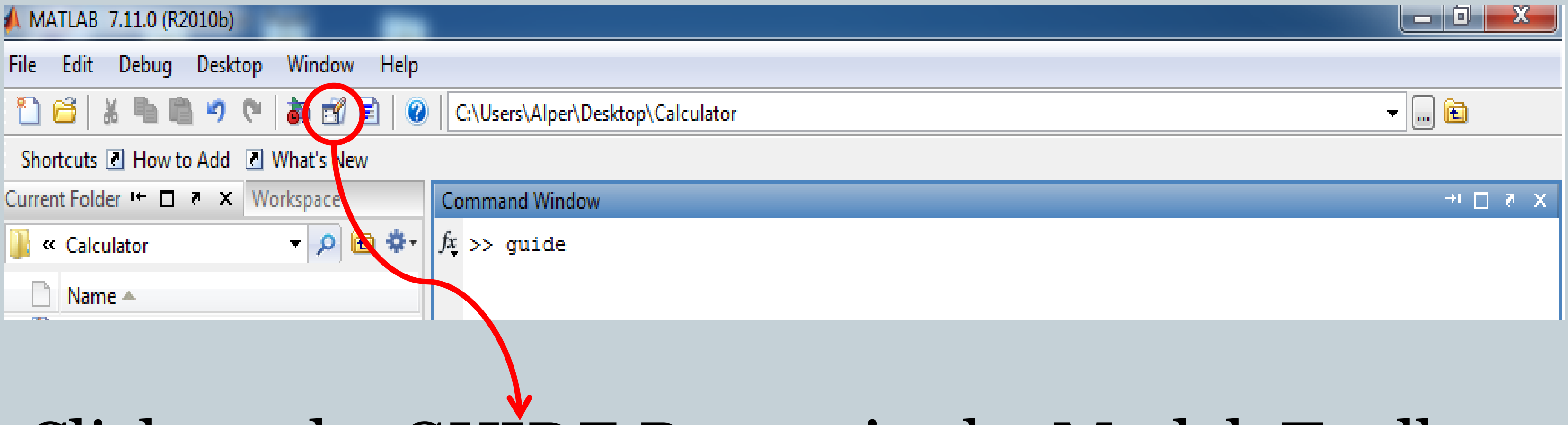
Keep in mind that GUI is just a make-up for the scripts that you have learned until now. Therefore, its language is no different than the usual scripts. However, the following steps can be utilized for guiding our path to GUI design.

- Form a figure that you think best fits to your problem.
- Assign this figure the functions that you have written for your program.
- Run your program.

Stages in GUI Design

There are two ways to open GUI in Matlab.

1. Write «guide» in Command Window.



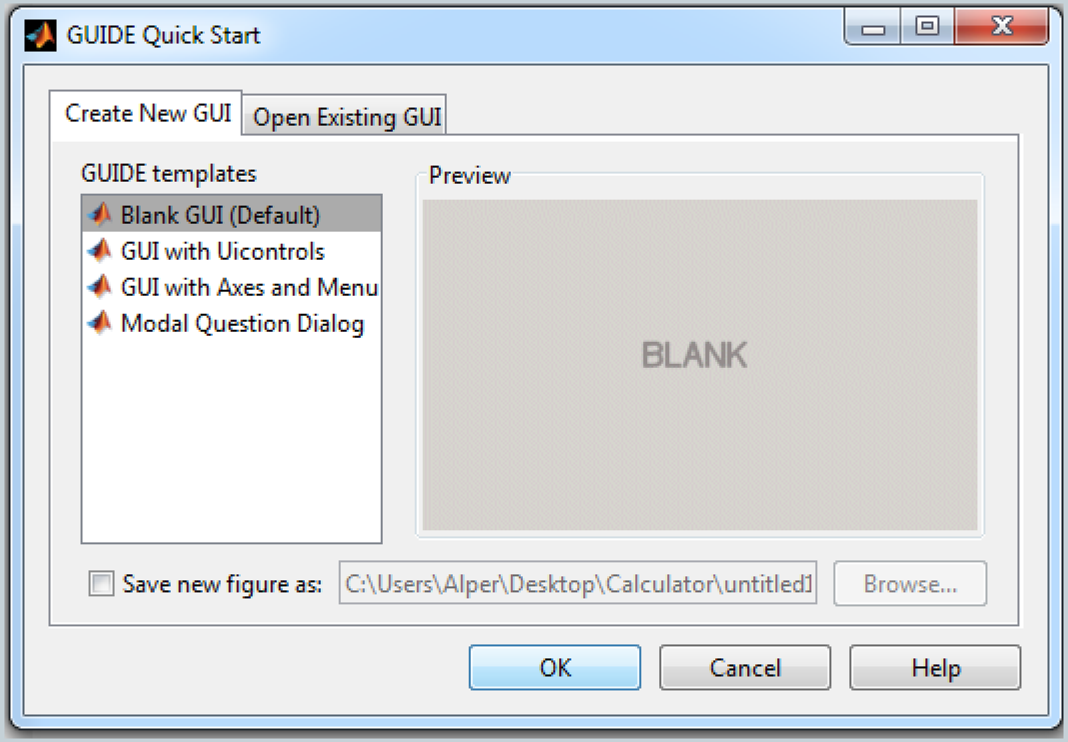
```
>> guide
```

2. Click on the GUIDE Button in the Matlab Toolbar.

Formation of Figure

Stages in GUI Design

After that the GUIDE Quick Start menu pops up.



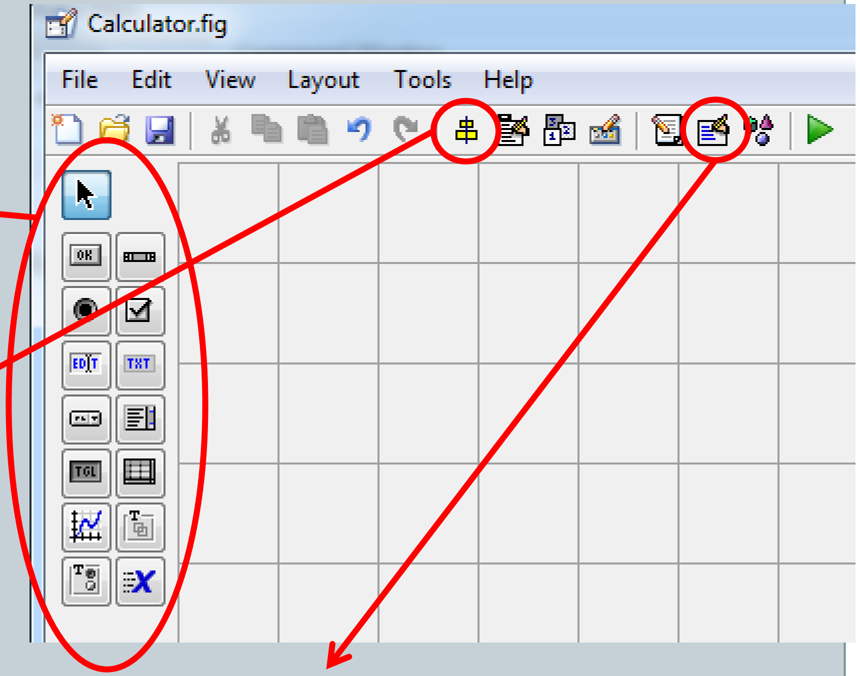
Save new figure as: C:\Users\Alper\Desktop\Calculator\untitled1

In this menu, select Blank GUI and press OK button.

Formation of Figure

Stages in GUI Design

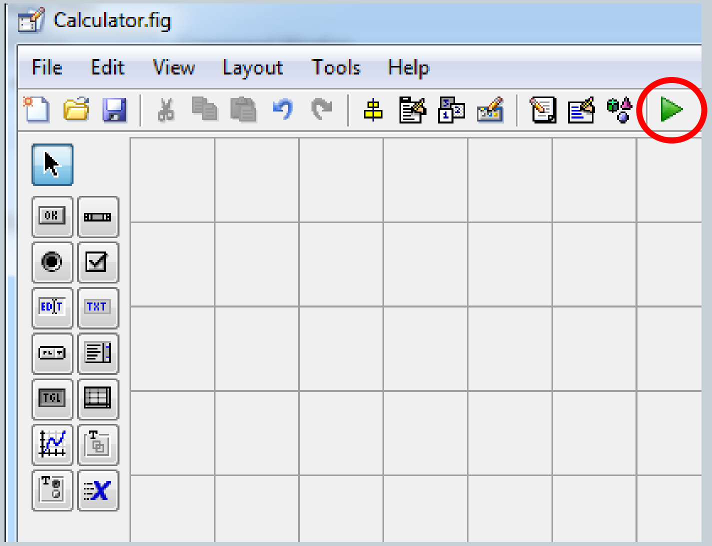
- A blank figure appears.
- Design your figure by using ***buttons on the left.***
- Align your buttons, text boxes, etc.) by using ***Align Object*** button.
- Edit your objects by utilizing ***Property Inspector*** button, or by right-clicking on your object and selecting Property Inspector.

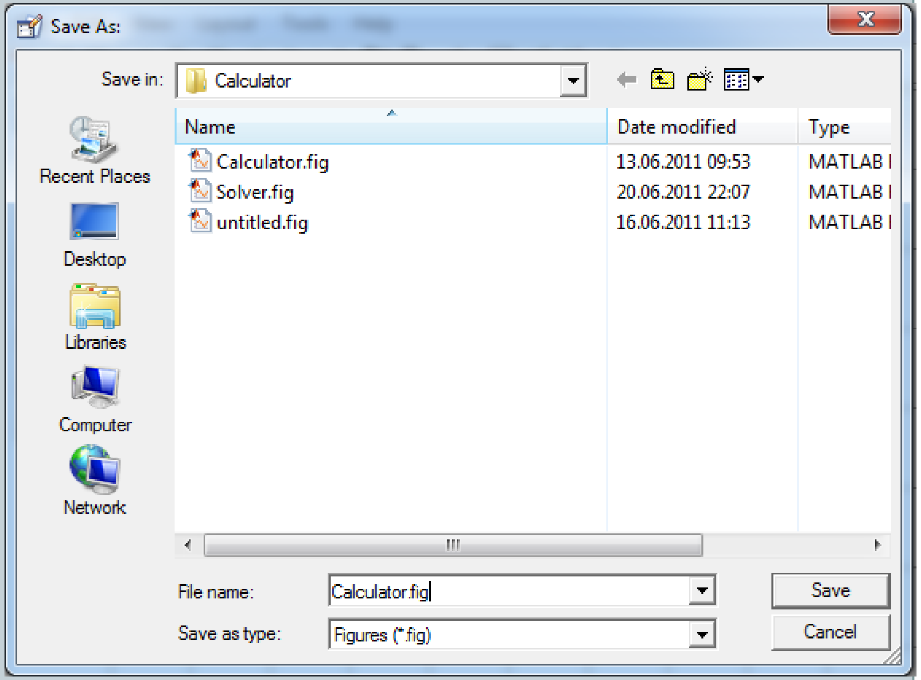


Formation of Figure

Stages in GUI Design

After you have finished your design click on the Run Figure button to save your figure and its M-file.





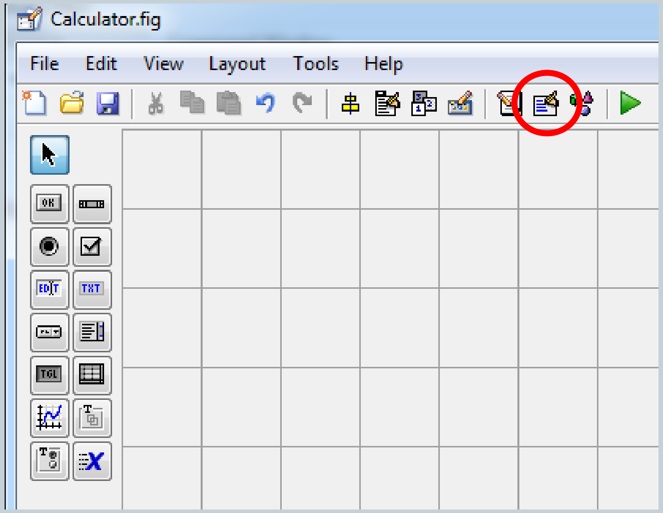
Name	Date modified	Type
Calculator.fig	13.06.2011 09:53	MATLAB I
Solver.fig	20.06.2011 22:07	MATLAB I
untitled.fig	16.06.2011 11:13	MATLAB I

File name: Calculator.fig
Save as type: Figures (*.fig)

Formation of Figure

Stages in GUI Design

In Matlab GUI, every object has its Tag (Name) present in the structure named «handles». If you want to call an object, you should use its Tag. Therefore, you have to give a Tag every object you add by utilizing ***Property Inspector*** Tool.



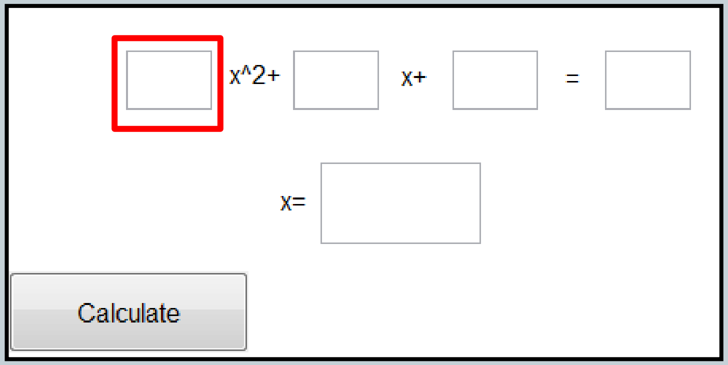
Assigning Functions to Objects

Stages in GUI Design

After tags of every object are determined, the properties of every object can be changed by utilizing the following built-in functions of MATLAB.

- get(handles.Tag,Property_Name)
- set(handles.Tag,Property_Name,Value)

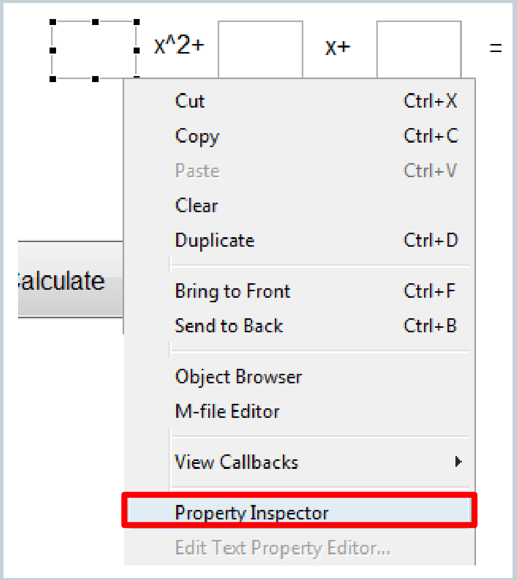
For example, let's change the Tag of the text box, then get its value to variable x and finally change its value as 95.



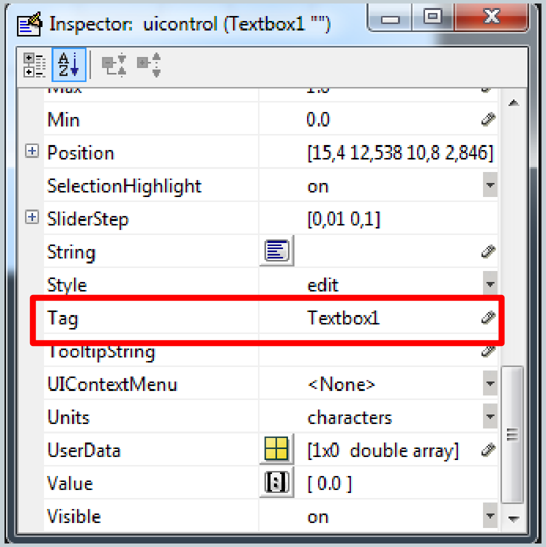
Assigning Functions to Objects

Stages in GUI Design

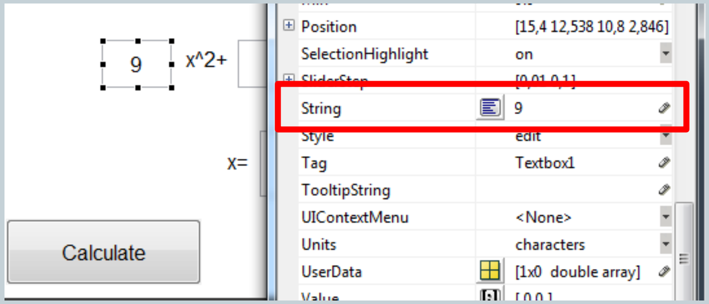
1. Select the text box and right click on it. Click on Property Inspector.



2. Change the Tag as Textbox1.



Min	0.0
Position	[15,4 12,538 10,8 2,846]
SelectionHighlight	on
SliderStep	[0,01 0,1]
String	
Style	edit
Tag	Textbox1
TooltipString	
UIContextMenu	<None>
Units	characters
UserData	[1x0 double array]
Value	[0.0]
Visible	on



Position	[15,4 12,538 10,8 2,846]
SelectionHighlight	on
String	9
Style	edit
Tag	Textbox1
TooltipString	
UIContextMenu	<None>
Units	characters
UserData	[1x0 double array]

Note that if you change the value of «String» from blank to 9, the textbox shows 9 inside.

Assigning Functions to Objects – Changing Tag

Stages in GUI Design

Now, let’s get its value to variable x by using the functions get.

«x=get(handles.TextBox1,'String')»

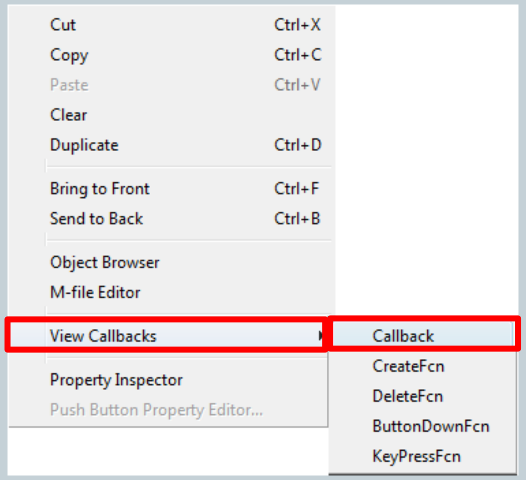
Finally, change its value as 95 by using function set.

«set(handles.Textbox1,'String','95')»

Assigning Functions to Objects – Value Operations

Stages in GUI Design

- Every object has also a callback function. For example, a callback function is activated when a button is clicked.
- To find the callback function of an object, right click on it and then select View Callbacks→Callbacks.



Assigning Functions to Objects – Finding Callbacks

Example - Calculator

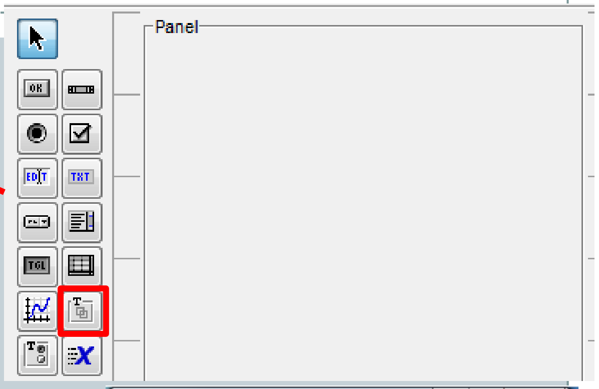
In this part, we will try to code a scientific calculator. Our aim is to create a calculator that is capable of

- Receiving arithmetic operations from user by means of buttons.
- It should keep the last answer in its memory. (Be careful. If no previous input exists, the answer is zero.)
- Instant stops should be prevented when syntax error exists. It should warn the user and give him/her a second chance.

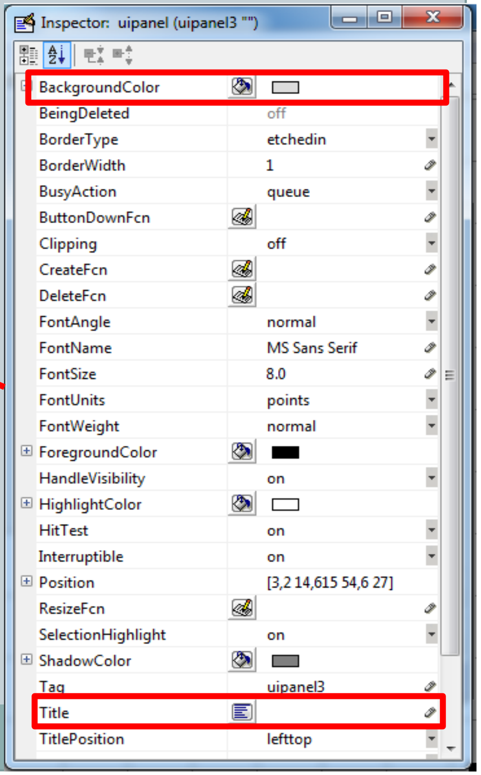
Problem Definition

Example - Calculator

1. Draw a panel for our calculator.



2. Delete the title «Panel» and give a grey colour (Backgroundcolor) to the panel by first selecting it and then opening Property Inspector.



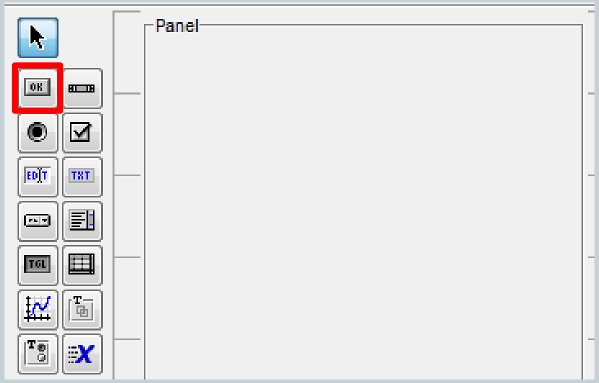
Inspector: uipanel (uipanel3 "")

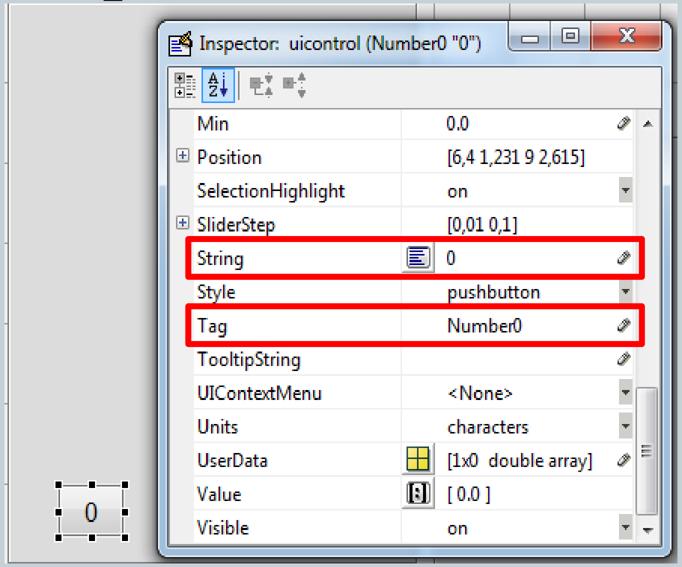
BackgroundColor	⬜
BeingDeleted	off
BorderType	etchedin
BorderWidth	1
BusyAction	queue
ButtonDownFcn	
Clipping	off
CreateFcn	
DeleteFcn	
FontAngle	normal
FontName	MS Sans Serif
FontSize	8.0
FontUnits	points
FontWeight	normal
ForegroundColor	■
HandleVisibility	on
HighlightColor	⬜
HitTest	on
Interruptible	on
Position	[3,2 14,615 54,6 27]
ResizeFcn	
SelectionHighlight	on
ShadowColor	▨
Tag	uipanel3
Title	
TitlePosition	lefttop

Designing the Calculator

Example - Calculator

3. Draw a push button and change its Tag as Number0 and its String as «0» from Property inspector.




Inspector: uicontrol (Number0 "0")

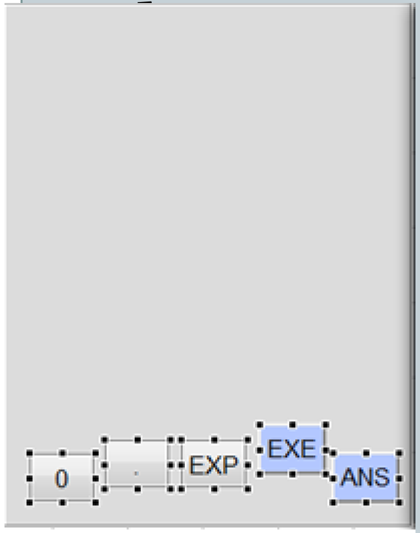
Min	0.0
Position	[6,4 1,231 9 2,615]
SelectionHighlight	on
SliderStep	[0,01 0,1]
String	0
Style	pushbutton
Tag	Number0
TooltipString	
UIContextMenu	<None>
Units	characters
UserData	[1x0 double array]
Value	[0.0]
Visible	on

4. Copy and paste this button five times. Give strings of other four buttons as «.», «EXP», «ANS» and «EXE». Define their tags as «Digit», «EXP», «ANS» and «EXE», respectively.

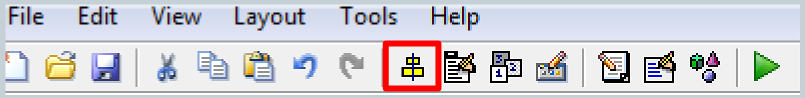
Designing the Calculator

Example - Calculator

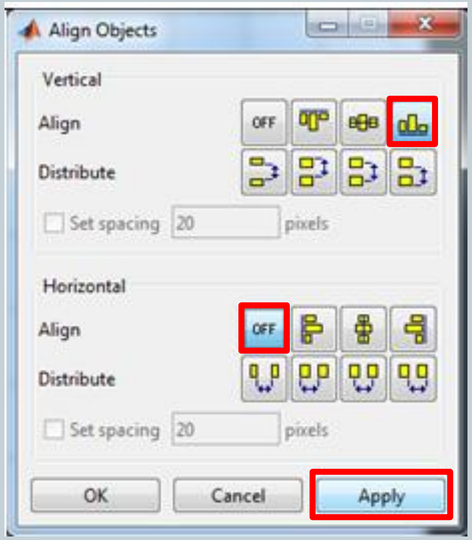
5. Align the buttons horizontally by using Align Objects



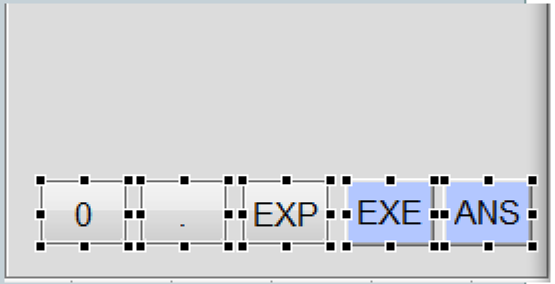
a. Select all of the buttons.



b. Click Align Objects button.



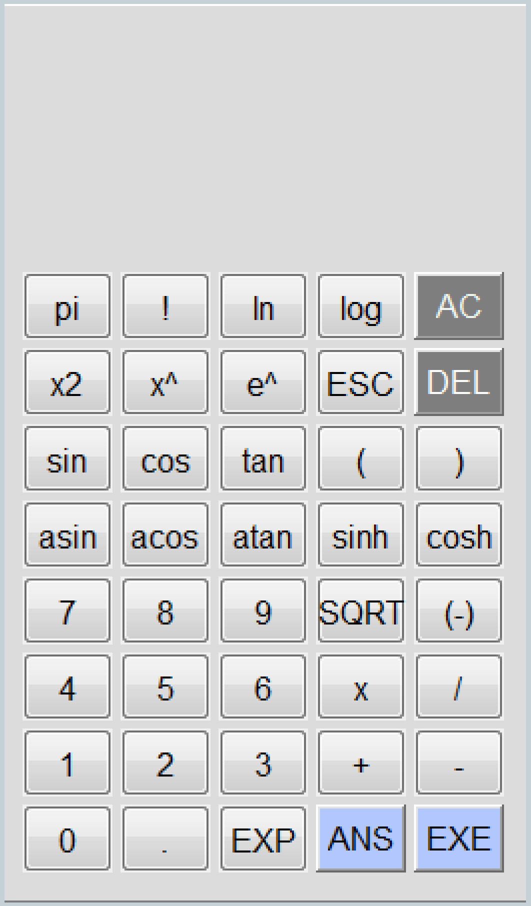
c. Select the above choices. Then, click Apply and close this window.



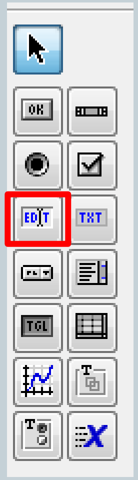
Designing the Calculator

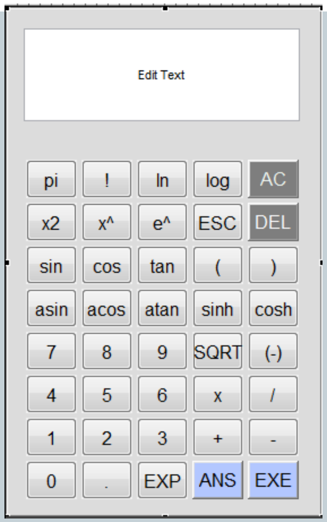
Example - Calculator

6. Draw all of the buttons given in the figure and assign them a tag and a corresponding string.



Example - Calculator

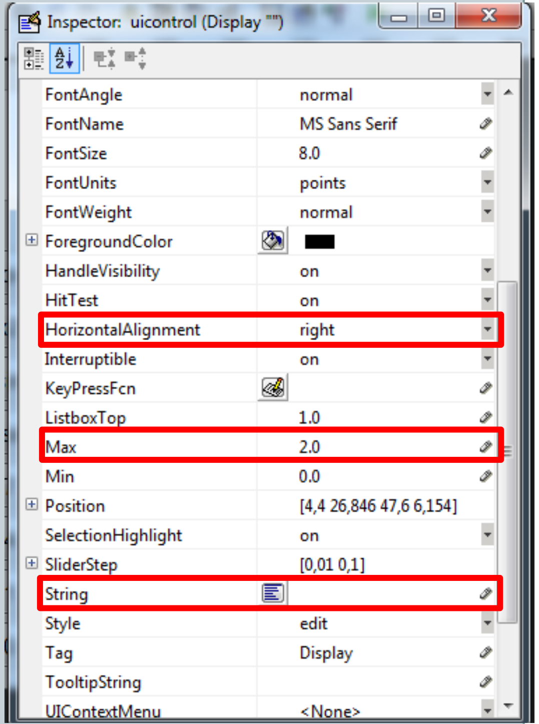




7. Now, draw a pretty large edit text box for the user input.

8. Delete «Edit Text» (String) of this box.

9. Align the text right (Horizontal Alignment) and in order to give the user the capability to write more than one lines make Max equal to 2.



Inspector: uicontrol (Display "")

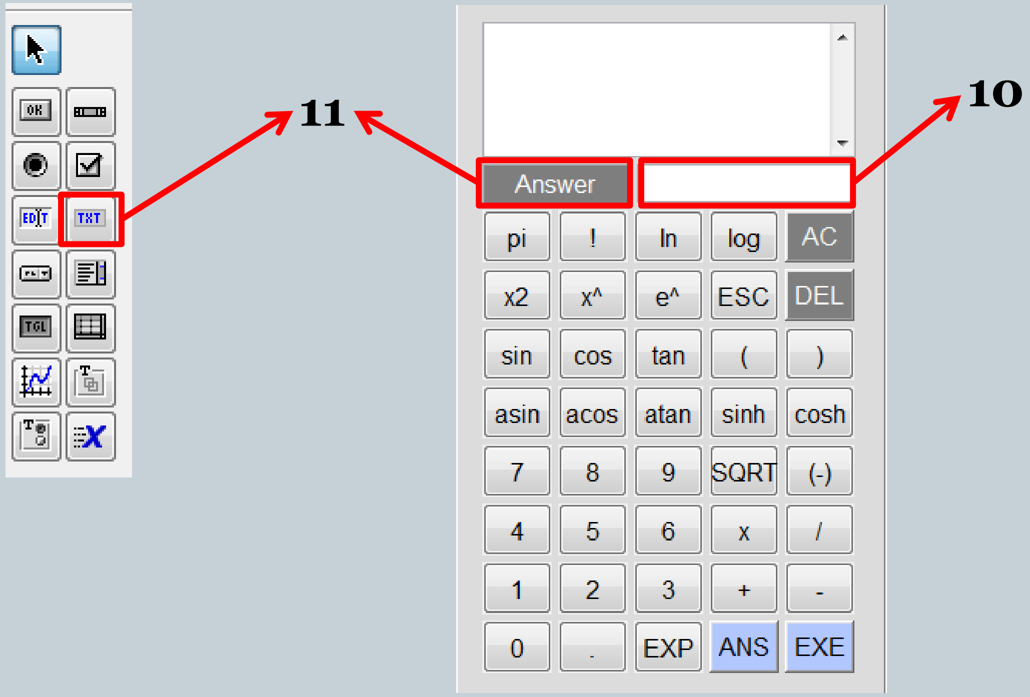
FontAngle	normal
FontName	MS Sans Serif
FontSize	8.0
FontUnits	points
FontWeight	normal
ForegroundColor	■
HandleVisibility	on
HitTest	on
HorizontalAlignment	right
Interruptible	on
KeyPressFcn	
ListboxTop	1.0
Max	2.0
Min	0.0
Position	[4,4 26,846 47,6 6,154]
SelectionHighlight	on
SliderStep	[0,01 0,1]
String	
Style	edit
Tag	Display
TooltipString	
UIContextMenu	<None>

Designing the Calculator

Example - Calculator

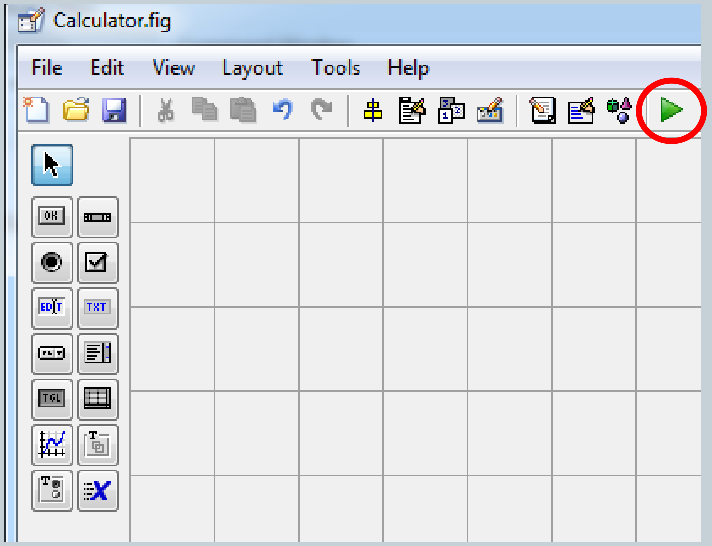
10. Draw one more edit text box for output.

11. Finally, draw one static text box to define the answer.



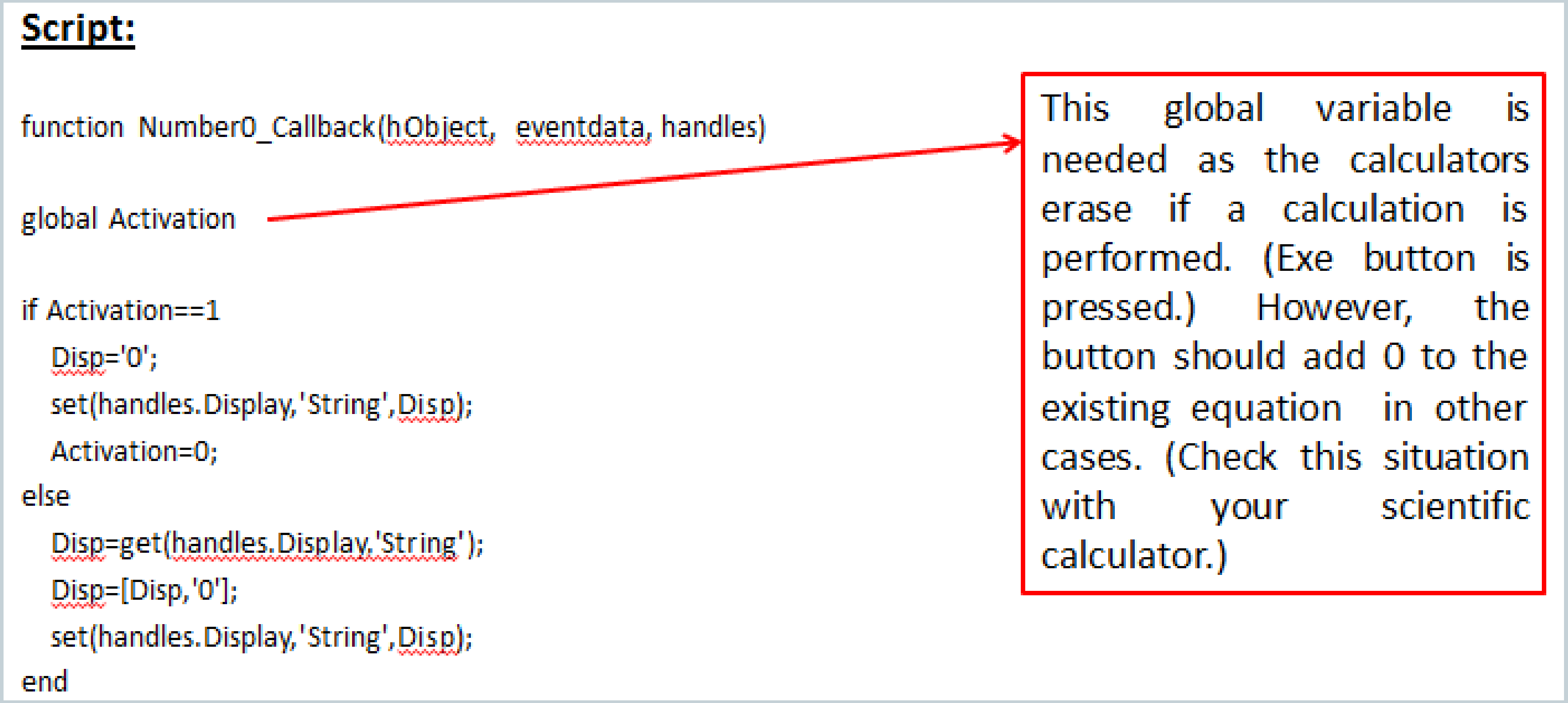
Example - Calculator

12. Run the figure to create figure (*.fig) and the code (*.m).



Example - Calculator

13. Add some script to all of the buttons. For example, if the user press «0», the display box (Step 7) should show «0» in addition to its previous content. This script should be added to the callback function of Number0.



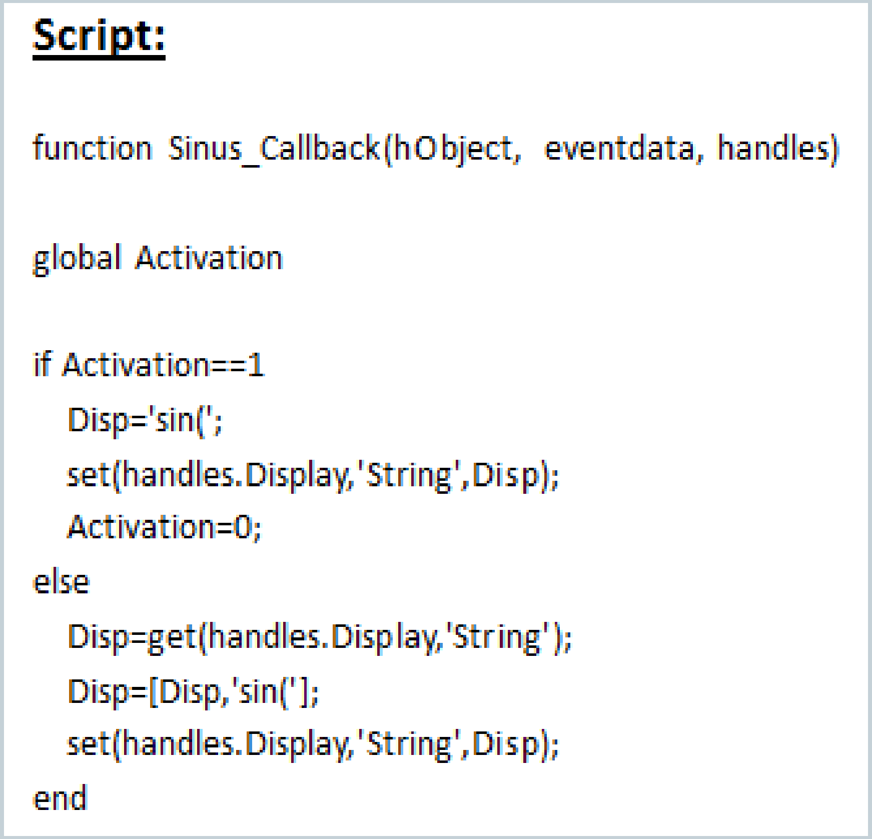
Script:

```
function  Number0_Callback(hObject,  eventdata, handles)

global Activation

if Activation==1
   Disp='0';
   set(handles.Display,'String',Disp);
   Activation=0;
else
   Disp=get(handles.Display,'String');
   Disp=[Disp,'0'];
   set(handles.Display,'String',Disp);
end
```

This global variable is needed as the calculators erase if a calculation is performed. (Exe button is pressed.) However, the button should add 0 to the existing equation in other cases. (Check this situation with your scientific calculator.)

Example - Calculator

14. The same code can be used for buttons other than (+, *, -, /, ANS, EXE, AC, ESC and DEL). For example for sin button, the following script works.

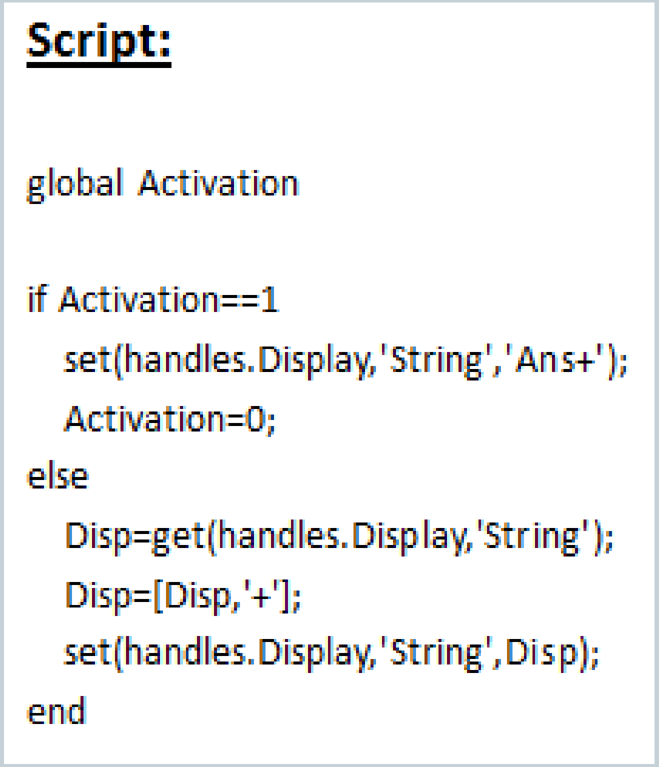


Script:

```
function  Sinus_Callback(hObject,  eventdata, handles)

global  Activation

if Activation==1
   Disp='sin(';
   set(handles.Display,'String',Disp);
   Activation=0;
else
   Disp=get(handles.Display,'String');
   Disp=[Disp,'sin('];
   set(handles.Display,'String',Disp);
end
```

Example - Calculator

15. If the user presses an operation button (+, -, * or /), the response should be the addition of this operation to the existing text when no calculations are performed whereas it should add «Ans+operation» in other cases. This situation is shown for Plus sign in the following code.

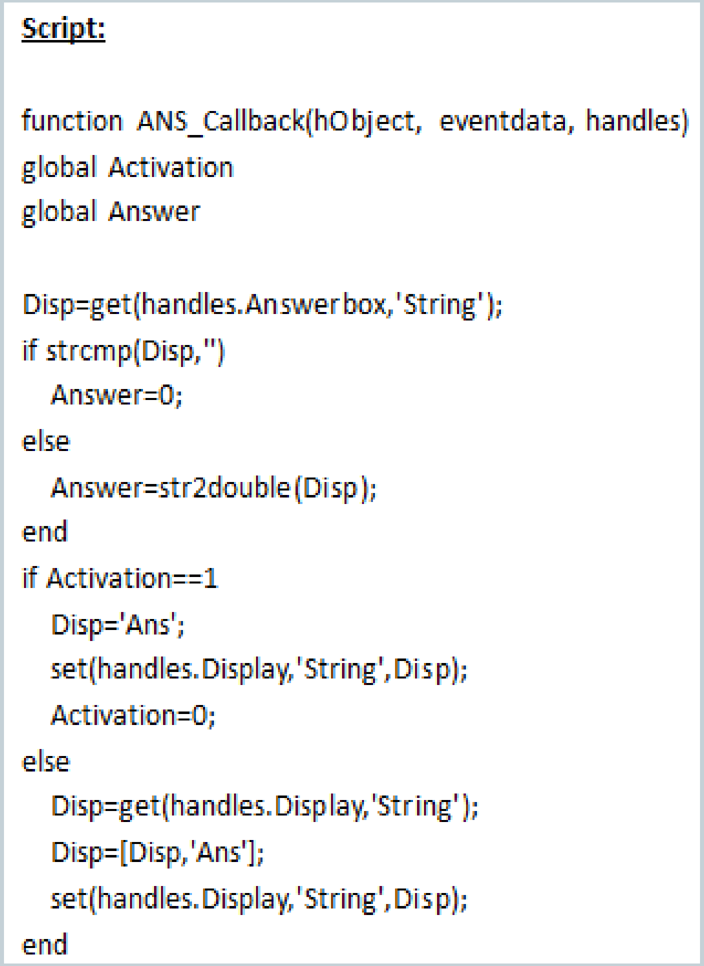


Script:

```
global Activation

if Activation==1
   set(handles.Display,'String','Ans+');
   Activation=0;
else
   Disp=get(handles.Display,'String');
   Disp=[Disp,'+'];
   set(handles.Display,'String',Disp);
end
```

Example - Calculator

16. ANS button should be capable of detecting the previous answer and record it.



Script:

```
function ANS_Callback(hObject, eventdata, handles)
global Activation
global Answer

Disp=get(handles.Answerbox,'String');
if strcmp(Disp,'')
   Answer=0;
else
   Answer=str2double(Disp);
end
if Activation==1
   Disp='Ans';
   set(handles.Display,'String',Disp);
   Activation=0;
else
   Disp=get(handles.Display,'String');
   Disp=[Disp,'Ans'];
   set(handles.Display,'String',Disp);
end
```

Writing Codes

Example - Calculator

17. AC button should be reset the calculator. In other words, it should delete the display and answer boxes and reset the activation status.

Script:

```
function AC_Callback(hObject, eventdata, handles)

global Activation

Activation=0;

Disp=[];
set(handles.Display,'String',Disp);
set(handles.Answerbox,'String',Disp);
```

Writing Codes

Example - Calculator

18. ESC button should be active only if a syntax error has occured. Moreover, it should reset the activation and bring the previously entered equation.

Script:

```
function  ESC_Callback(hObject,  eventdata, handles)

global Activation
global Answer
global Equation1
global Error

if Error==1
   Activation=0;
   set(handles.Display,'String',Equation1);
   set(handles.Answerbox,'String',Answer);
end
```

Example - Calculator

18. DEL button should delete one character when pressed. However, the code can be improved to prevent syntax error and to ease the user control. To do this, the code should be capable of detecting the last word and if it is a special one like sin, cos, etc. it should delete sin, cos, etc. completely.

```
function DEL_Callback(hObject, eventdata, handles)

Disp=get(handles.Display,'String');

if isempty(Disp)==1
  %Do nothing
elseif strcmp(Disp(length(Disp)),'(')
  Disp=Disp(1:length(Disp)-1);
else
  if length(Disp)>=4
    if strcmp(Disp((length(Disp)-3):length(Disp)),'asin')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'acos')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'atan')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'sqrt')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'sinh')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'cosh')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-2):length(Disp)),'sin')
      Disp=Disp(1:length(Disp)-3);
    elseif strcmp(Disp((length(Disp)-2):length(Disp)),'cos')
      Disp=Disp(1:length(Disp)-3);
    elseif strcmp(Disp((length(Disp)-2):length(Disp)),'tan')
      Disp=Disp(1:length(Disp)-3);
    elseif strcmp(Disp((length(Disp)-2):length(Disp)),'Ans')
      Disp=Disp(1:length(Disp)-3);
    elseif strcmp(Disp((length(Disp)-2):length(Disp)),'log')
      Disp=Disp(1:length(Disp)-3);
    elseif strcmp(Disp((length(Disp)-1):length(Disp)),'pi')
      Disp=Disp(1:length(Disp)-2);
    elseif strcmp(Disp((length(Disp)-1):length(Disp)),'ln')
      Disp=Disp(1:length(Disp)-2);
    else
      Disp=Disp(1:length(Disp)-1);
    end
  elseif length(Disp)==4
    if strcmp(Disp((length(Disp)-3):length(Disp)),'asin')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'acos')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'atan')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'sqrt')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'sinh')
      Disp=Disp(1:length(Disp)-4);
    elseif strcmp(Disp((length(Disp)-3):length(Disp)),'cosh')
      Disp=Disp(1:length(Disp)-4);
    else
      Disp=Disp(1:length(Disp)-1);
    end
  elseif length(Disp)==3
    if strcmp(Disp((length(Disp)-2):length(Disp)),'sin')
      Disp=Disp(1:length(Disp)-3);
    elseif strcmp(Disp((length(Disp)-2):length(Disp)),'cos')
      Disp=Disp(1:length(Disp)-3);
    elseif strcmp(Disp((length(Disp)-2):length(Disp)),'tan')
      Disp=Disp(1:length(Disp)-3);
    elseif strcmp(Disp((length(Disp)-2):length(Disp)),'Ans')
      Disp=Disp(1:length(Disp)-3);
    elseif strcmp(Disp((length(Disp)-2):length(Disp)),'log')
      Disp=Disp(1:length(Disp)-3);
    else
      Disp=Disp(1:length(Disp)-1);
    end
  elseif length(Disp)==2
    if strcmp(Disp((length(Disp)-1):length(Disp)),'pi')
      Disp=Disp(1:length(Disp)-2);
    elseif strcmp(Disp((length(Disp)-1):length(Disp)),'ln')
      Disp=Disp(1:length(Disp)-2);
    else
      Disp=Disp(1:length(Disp)-1);
    end
  else
    Disp=Disp(1:length(Disp)-1);
  end
end

set(handles.Display,'String',Disp);
```

Writing Codes

Example - Calculator

19. EXE button should detect the syntax error and do not stop. (Hint: use Try and catch.) Moreover, this button should transform the eqaution given by the user to a Matlab compatible format. In other words, The exclamation mark (!) has no meaning for Matlab but the user presses this button for the factorial operation. Therefore, this mark should be removed and the built-in function factorial has to be added into the equation.

For example, the user may give the following equation.

5*sin(10)+9!

If the above equation is given to Matlab, it causes an error. Thus, the given equation should be transformed to the following one.

5*sin(10)+factorial(9)

Script:

```
function EXE_Callback(hObject, eventdata, handles)


global Activation
global Answer
global Error
global Equation1

Activation=1;

Equation1=get(handles.Display,'String');

try
   Equation=get(handles.Display,'String');

   if size(findstr(Equation,'Ans'),2)>=1
      Equation=strrep(Equation,'Ans',num2str(Answer));
   end

   if size(findstr(Equation,'e^'),2)>=1
      Equation=strrep(Equation,'e','exp(1)');
   elseif size(findstr(Equation,'log('),2)>=1
      Equation=strrep(Equation,'log(','log10(');
   elseif size(findstr(Equation,'ln('),2)>=1
      Equation=strrep(Equation,'ln(','log(');
   elseif size(findstr(Equation,'!'),2)>=1
      [Start, Location]=Locationfinder(Equation);
      Values=cell(size(Location,2),1);
      for i=1:1:size(Location,2)
         Values{i,1}=Equation(Start(1,i)+1:Location(1,i)-1);
      end
      for i=1:1:size(Location,2)
         Equation=strrep(Equation,[Values{i,1} '!'],['factorial(' Values{i,1} ')']);
      end
   end
   Answer=eval(Equation);
   set(handles.Answerbox,'String',num2str(Answer));
catch
   Error=1;
   set(handles.Display,'String','Syntax Error. Press ESC to return back and check your entry.');
end
```