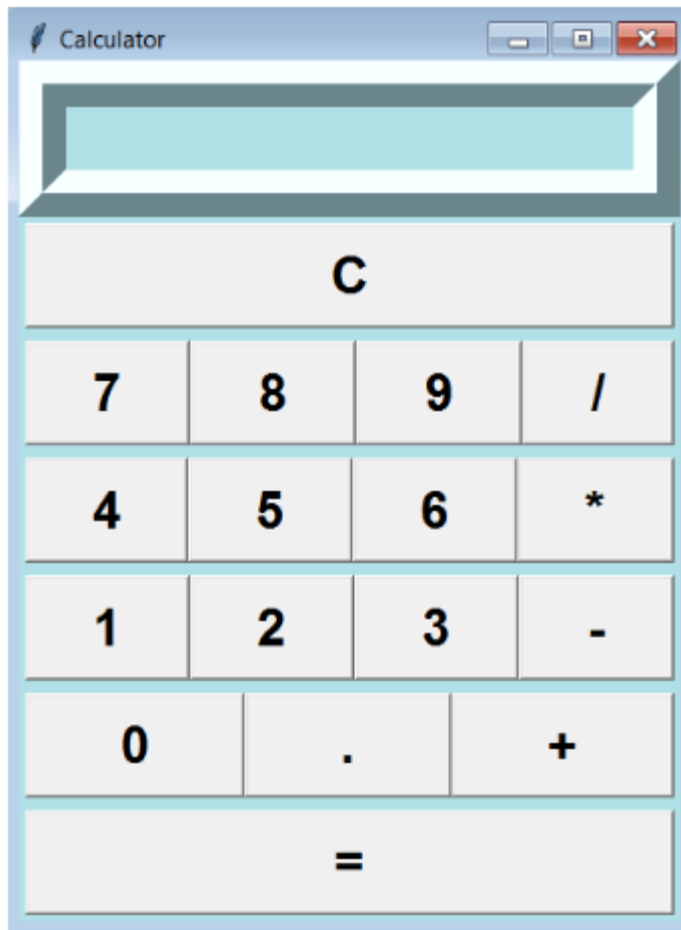


Laboratory Activity No. 11	
The Grid Manager	
Course Code: CPE103	Program: BSCPE
Course Title: Object-Oriented Programming	Date Performed: April 5, 2025
Section: 1--A	Date Submitted: April 5, 2025
Name: Villacin, Justine R.	Instructor: Maria Rizette H. Sayo
1. Objective(s):	
This activity aims to familiarize students on how to implement geometry manager	
2. Intended Learning Outcomes (ILOs):	
The students should be able to:	
2.1 Identify the main components in a GUI Application	
2.2 Create a simple GUI Application using Grid manager	
3. Discussion:	
<p>A Graphical User Interface (GUI) application is a program that the user can interact with through graphics (windows, buttons, text fields, checkboxes, images, icons, etc..) such as the Desktop GUI of Windows OS by using a mouse and keyboard unlike with a Command-line program or Terminal program that support keyboard inputs only.</p> <p>Geometry managers are tools used to place widgets on the screen. There are three geometry managers available in tkinter—grid, pack, and place. The place manager provides complete control in the positioning of widgets, but is complicated to program</p> <p>Grids</p> <ul style="list-style-type: none"> A grid is an imaginary rectangle containing horizontal and vertical lines that subdivide it into rectangles called cells. The first row of cells is referred to as row 0, the second row is referred to as row1, and so on. Similarly, the first column of cells is referred to as column 0, the second column of cells is referred to as column 1, and so on. Each cell is identified by its row and column numbers. 	
4. Materials and Equipment:	
<p>Desktop Computer with Pycharm</p> <p>Windows Operating System</p>	
5. Procedure:	

General Instruction:

1. Redesign the interface of the standard calculator using grid () method:



2. Run the program and observe the output when the button is clicked.

6. Supplementary Activity:

1. Make a calculator program that can compute perform the Arithmetic operations as well as exponential operation, sin, cosine math functions as well clearing using the C button and/or clear from a menu bar.
2. Use Geometry manager grid()
3. Use bind () or command parameter in associating event to callback a function.

Questions

1. How do you configure rows and columns in PyCharm when using Tkinter's grid() manager?
 - *You can get things in the window lined up by assigning each widget a 'row' and 'column' number through 'grid()'. It keeps all things tidy and in their proper place on the screen.*
2. Why do widgets sometimes disappear when using grid() in PyCharm, and how can you fix it?
 - *Sometimes, my widgets vanished when I combine 'pack()' and 'grid()' or fail to set the proper row/column, nightmare. That was corrected by employing only 'grid()' in all of my frames and double-checking widget positions.*
3. How can message boxes be used to provide a better User Experience or how can message boxes be used to make a GUI Application more user-friendly? How can you align widgets across multiple frames using grid() in PyCharm?
 - *Message boxes do serve a purpose in that they inform users what is occurring, such as displaying errors or prompting them to confirm. In order to put things into separate frames synchronously, I applied 'grid()' on both frames using the same row and column configurations and applied sticky and padding to get it to appear pretty.*

7. Conclusion:

Based on this activity, Tkinter's grid() manager is an effective method of placing widgets into a neat and organized layout. grid() assists in placing buttons and input fields precisely where required by defining row and column values precisely. grid() also provides flexibility through colspan, padx, pady, and sticky options to make the interface more aesthetically pleasing and user-friendly. By employing grid() throughout and with precision, both calculators possess clean, working looks with no overlap or misplaced widgets—demonstrating just how strong and versatile grid() can be in actual GUI programs.

8. Assessment Rubric: