Laboratory Activity No. 11	
The Grid Manager	
Course Code: CPE103	**Program:** BSCPE
Course Title: Object-Oriented Programming	**Date Performed:05/04/2025**
Section:1A	**Date Submitted:05/04/2025**
Name:Jalilah M. Disomnong	**Instructor: Engr. 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:

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.

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

Grids

- 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.

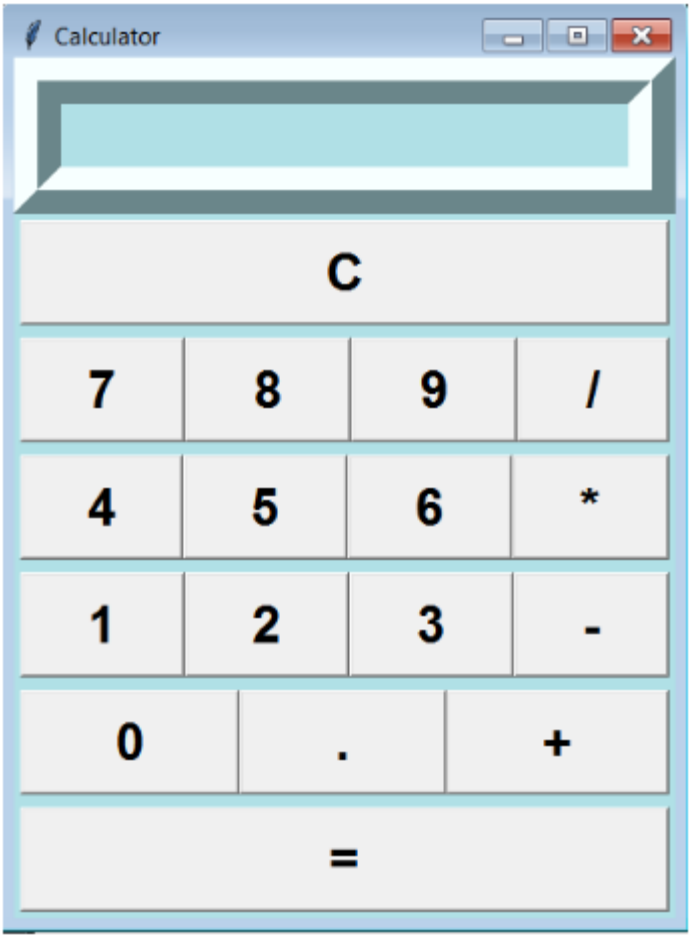
4. Materials and Equipment:

Desktop Computer with Pycharm
Windows Operating System

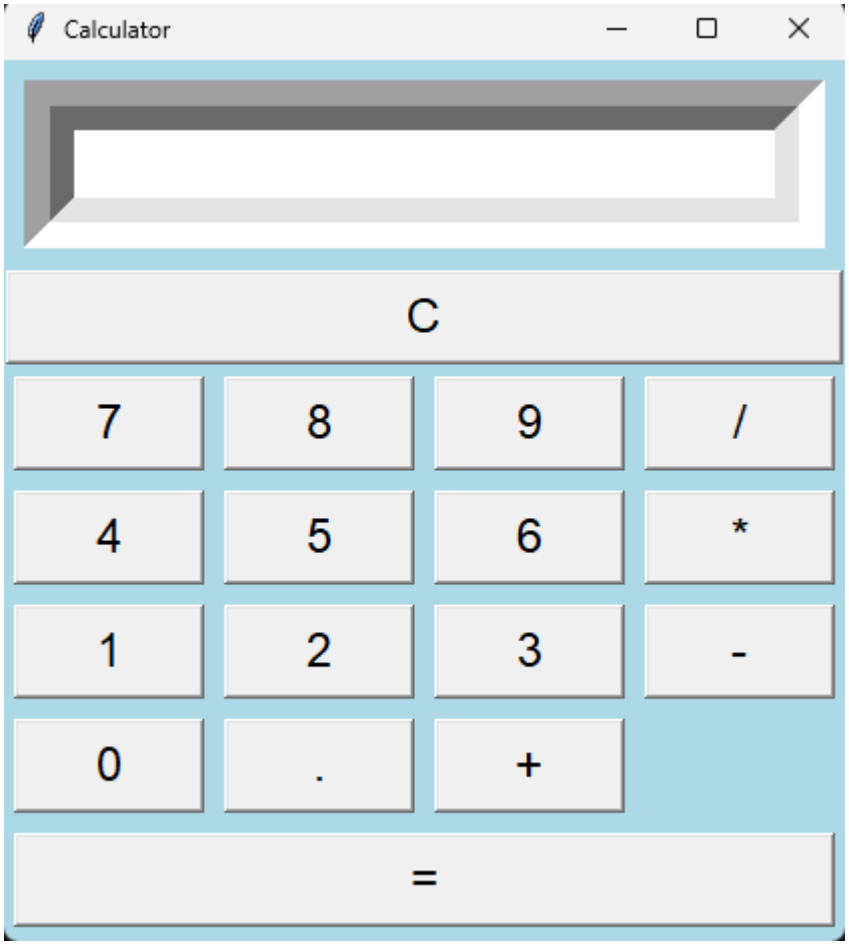
5. Procedure:

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.

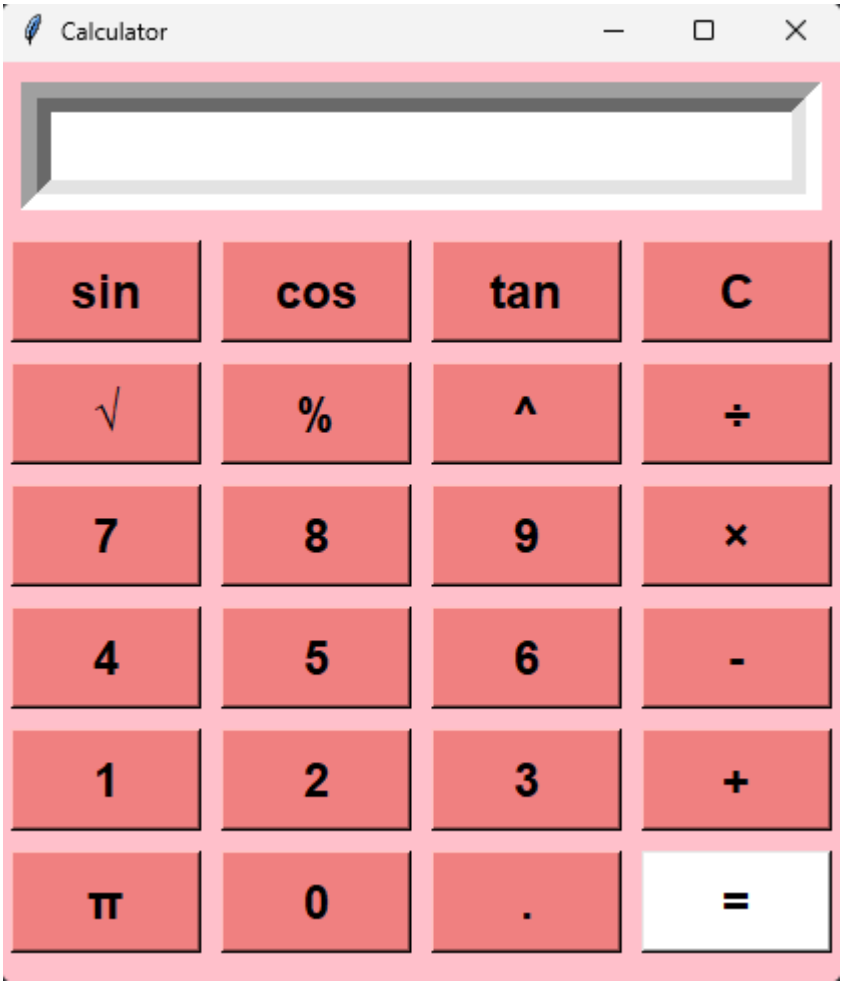


Redesign: Please refer to the Calculator.py for source code

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.



Please refer to the supplementary_lab11 file for the source code

Questions

1. How do you configure rows and columns in PyCharm when using Tkinter's grid() manager?
   To configure rows and columns in PyCharm using Tkinter's grid() manager, use the rowconfigure() and columnconfigure() methods. Set the weight of rows and columns to ensure they expand proportionally when the window is resized

2. Why do widgets sometimes disappear when using grid() in PyCharm, and how can you fix it?
   Widgets sometimes disappear when using grid() in PyCharm due to conflicts with other geometry managers like pack() or place(). Tkinter does not allow mixing different geometry managers within the same container. To fix this issue, ensure you are consistently using the grid() manager for all widgets within the same container.

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 improve user experience by providing clear feedback, warnings, and confirmations. They alert users about errors, confirm actions, and provide important information. To align widgets across multiple frames using grid(), configure the rows and columns of each frame with the same weights. This ensures widgets align properly across frames.

7. Conclusion:

In this laboratory activity, we focused on implementing the grid() geometry manager in Tkinter to create a simple GUI application. I learned how to effectively use the grid() geometry manager in Tkinter to create a simple GUI application. By configuring rows and columns with the rowconfigure() and columnconfigure() methods, I ensured that the layout was responsive and adjusted well to different window sizes. This activity provided a solid foundation for developing more complex GUI applications using Tkinter.

8. Assessment Rubric: