Laboratory Activity No. 11	
The Grid Manager	
Course Code: CPE103	**Program:** BSCPE
Course Title: Object-Oriented Programming	**Date Performed:** 04/05/2025
Section: 1-A	**Date Submitted:** 04/05/2025
Name: BRON, JHUSTINE A.	**Instructor:** ENGR. MARIA RIZETTE 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.

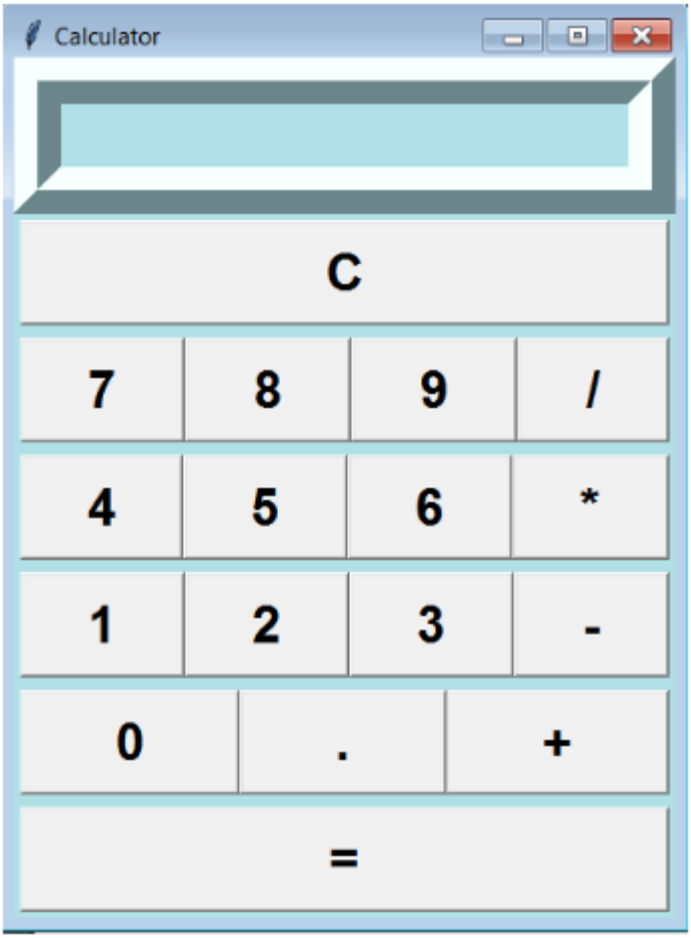
4. Materials and Equipment:

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.

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?

   In PyCharm, when using Tkinter's grid() manager, you can configure rows and columns by specifying row and column numbers for each widget with grid(row=x, column=y). You can also use rowspan and columnspan to make widgets span across multiple rows or columns.

2. Why do widgets sometimes disappear when using grid() in PyCharm, and how can you fix it?

   Widgets may disappear if you don't properly set the row and column sizes or if you forget to call pack() or grid() in the right order. To fix this, ensure you're calling grid() for all widgets and set sticky to make widgets stretch in their cells. Also, sometimes using root.grid_rowconfigure() or root.grid_columnconfigure() helps.

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 help by showing notifications or error messages, making it clearer for users. They can alert users about important actions like successful operations or input mistakes. For aligning widgets across multiple frames, you can use grid() by specifying consistent row and column values for each widget, ensuring they line up across all frames.

7. Conclusion:

In conclusion, when using Tkinter's grid() manager in PyCharm, I can easily organize and arrange widgets in my GUI by configuring rows and columns. If widgets disappear, it's usually because the row/column sizes aren't set properly or the grid() calls are out of order. Message boxes help improve the user experience by giving feedback or alerts about important actions or mistakes. To align widgets across frames, I just need to keep the row and column values consistent. By following these tips, I can make my app more intuitive and user-friendly.

8. Assessment Rubric: