Laboratory Activity No. 11	
The Grid Manager	
Course Code: CPE103	**Program:** BSCPE
Course Title: Object-Oriented Programming	**Date Performed:** 05/04/2025
Section: 1-A	**Date Submitted:** 05/04/2025
Name: Delinia, Filjohn B.	**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.

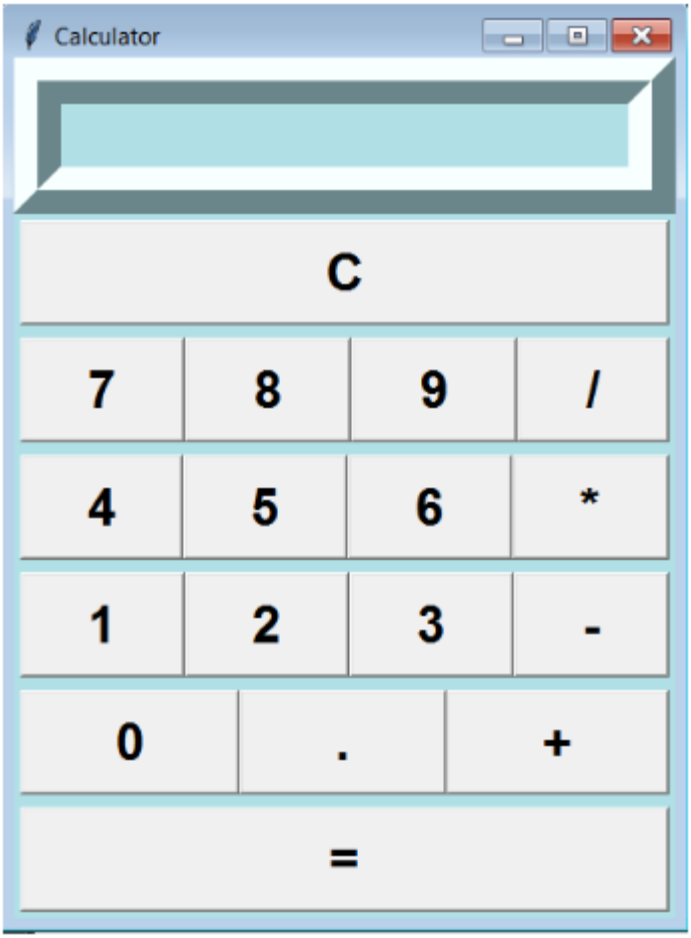
4. Materials and Equipment:

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?
   When we are using Tkinter's grid() manager, we configure rows and columns by specifying the row and column parameters for each widget , and we can control the resizing behavior by configuring row and column weights grid_rowconfigure() and grid_columnconfigure methods

2. Why do widgets sometimes disappear when using grid() in PyCharm, and how can you fix it?
   Widgets may disappear when I use grid() in PyCharm if I mix it with pack(), forget to specify row/column positions, or don't configure row/column weights properly. I can fix this by sticking to one layout manager per container and ensuring correct grid configuration.

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 me improve the user experience in my GUI by providing useful alerts, warnings, or confirmations, which makes the application more user-friendly. To align widgets across multiple frames, I use consistent row and column values in grid() to ensure everything is laid out correctly.

7. Conclusion:

In conclusion, when I use Tkinter's grid() manage, I need to carefully configure the rows and columns, manage widget placements, and adjust row and column weights for proper resizing behavior. To prevent issues like disappearing widgets, I should avoid mixing layout managers and making sure all grid configurations are correct. Additionally, message boxes help me improve the user experience by providing clear feedback, while aligning widgets across multiple frames is easy when I use consistent grid settings.

8. Assessment Rubric: