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

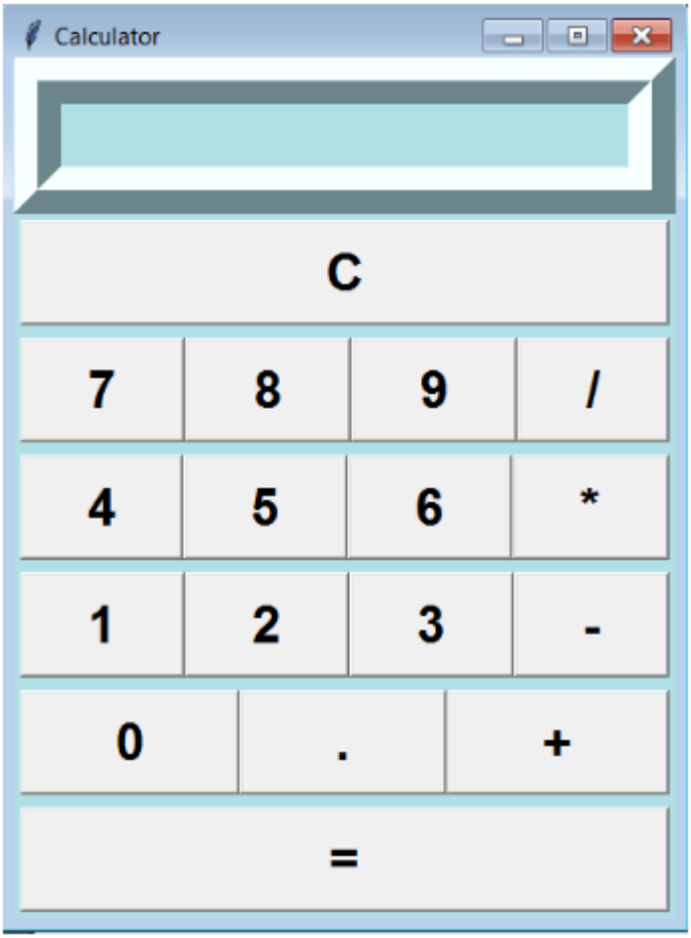
4. Materials and Equipment:

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.

For the output please refer to this link: CPE-103-OOP-1-A/LAB ACT 11/SUPPLEMENTARY 11.py at main · Lewis-Clark-Palmes/CPE-103-OOP-1-A

Questions

1. How do you configure rows and columns in PyCharm when using Tkinter's grid() manager?
   In Tkinter, the grid() manager allows configuring rows and columns using the row=() and column= () methods. These methods define how widgets expand and adjust within a window when resized, improving flexibility in layout design. While its easy to understand it takes time to apply since you may face many errors or have a hard time putting a button in the desired place

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

   Widgets may disappear when using grid() due to improper placement, incorrect parent widget configuration, or missing the mainloop() call. Ensuring correct widget assignments, visible parents, and running mainloop() prevents these issues and maintains a functional GUI.

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 enhance user experience by providing instant notifications, alerts, or confirmations. They help users receive feedback or warnings, making applications more interactive. The grid() manager also ensures widgets are aligned properly across multiple frames using parameters like sticky and rowspan, allowing structured and organized layouts that contribute to responsiveness and effective user interaction.

7. Conclusion:

The grid() geometry manager in Tkinter provides a structured way to organize widgets within a GUI application. Proper configuration of rows and columns ensures flexibility and responsiveness, while addressing widget visibility issues prevents unintended disappearing elements. Using message boxes enhances user interaction by providing clear notifications and feedback, making the application more intuitive. By effectively utilizing grid() and related functions, we students can create well-structured, user-friendly interfaces that improve the overall output and experience. Overall, the activity reinforced essential concepts in object-oriented programming for GUI development.

8. Assessment Rubric: