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 10, 2025
Name: Directo, Hannah Thea 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.

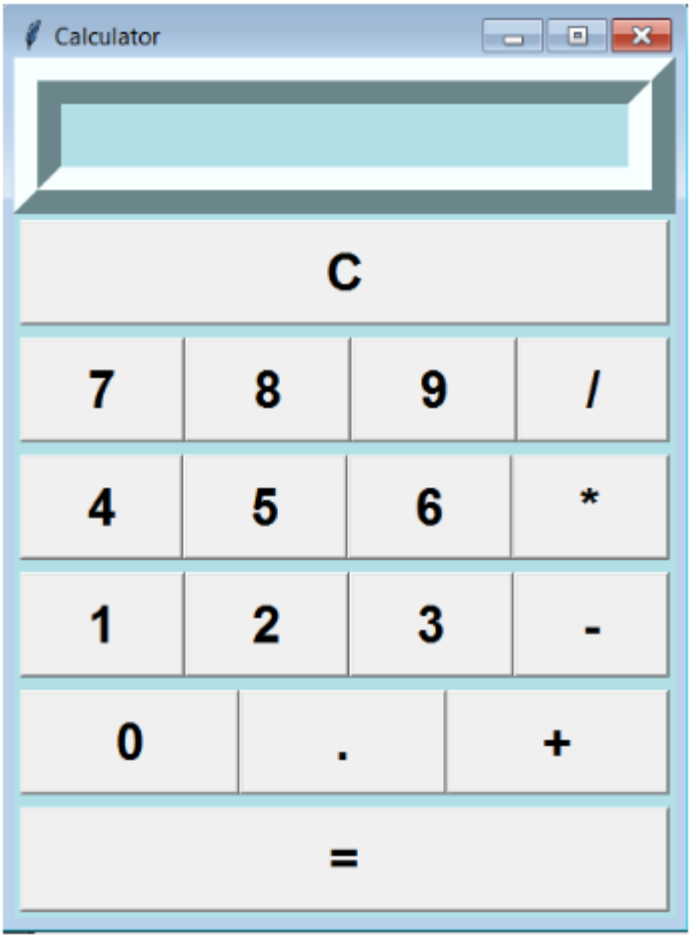
4. Materials and Equipment:

Desktop Computer with Pycharm
Windows Operating System

5. Procedure:

General Instruction:

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

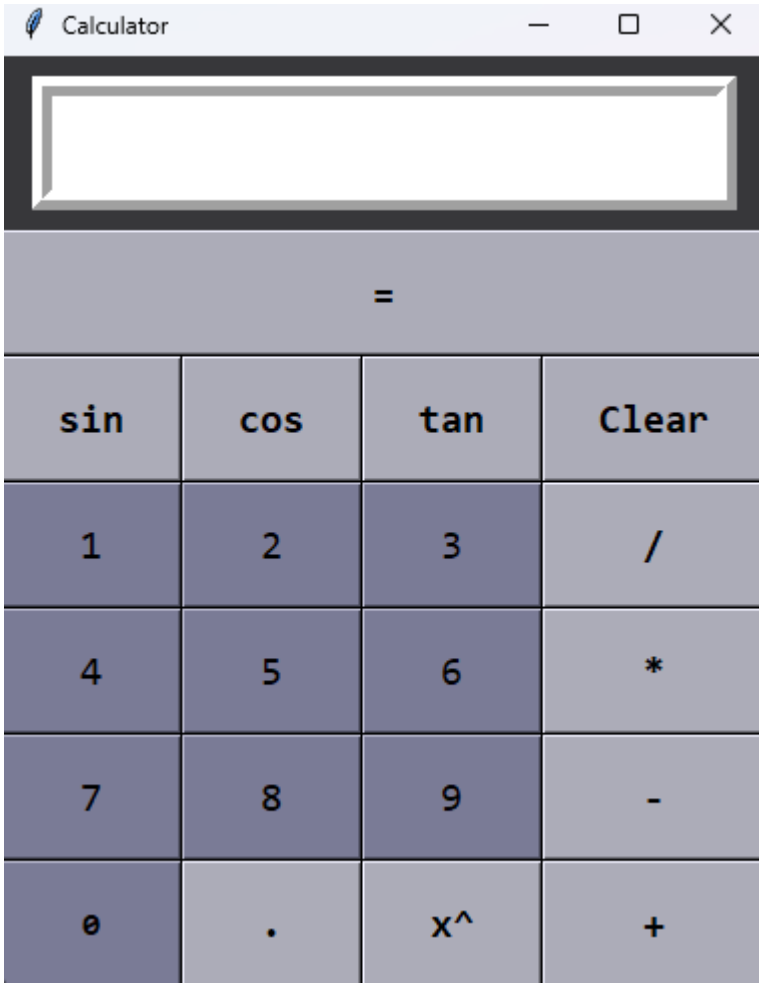


2. Run the program and observe the output when the button is clicked.

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 using the grid() layout in Tkinter, rows and columns can be setup using grid_rowconfigure() and grid_columnconfigure() on the main window or frame. This helps to control how the space is when the window gets bigger or smaller.

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

   Widgets can sometimes disappear when using grid() in PyCharm if the parent container is not configure or displayed properly. This happens when the frame containing the widget is not added to the main window using pack(), grid(), or place(). To fix this, ensure all containers and widgets are properly gridded or packed, and double-check the layout to avoid conflicts or misplacement.

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 make your app easier to use by showing pop-up windows with messages, warnings, or questions. They guide users through the application by providing interactive prompts confirm actions, or notify them of important information. To make widgets line up nicely across different frames, use the same row and column number, and use options like sticky to keeps things in place. You can also add space around widgets using padx and pady.

7. Conclusion:

In this laboratory activity, I learned how to create a simple calculator using Python's Tkinter library and the `grid()` layout manager. I practiced placing buttons and other elements in specific rows and columns to design the layout. This helped me understand how to organize parts of a GUI using the grid system. I also learned how to add functions to buttons using the `'command'` parameter and how to use message boxes to make the program easier to use. The activity helped me become more familiar with building user interfaces and how layout managers work in Python. Overall, this lab improved my skills in designing and coding simple GUI applications.

8. Assessment Rubric: