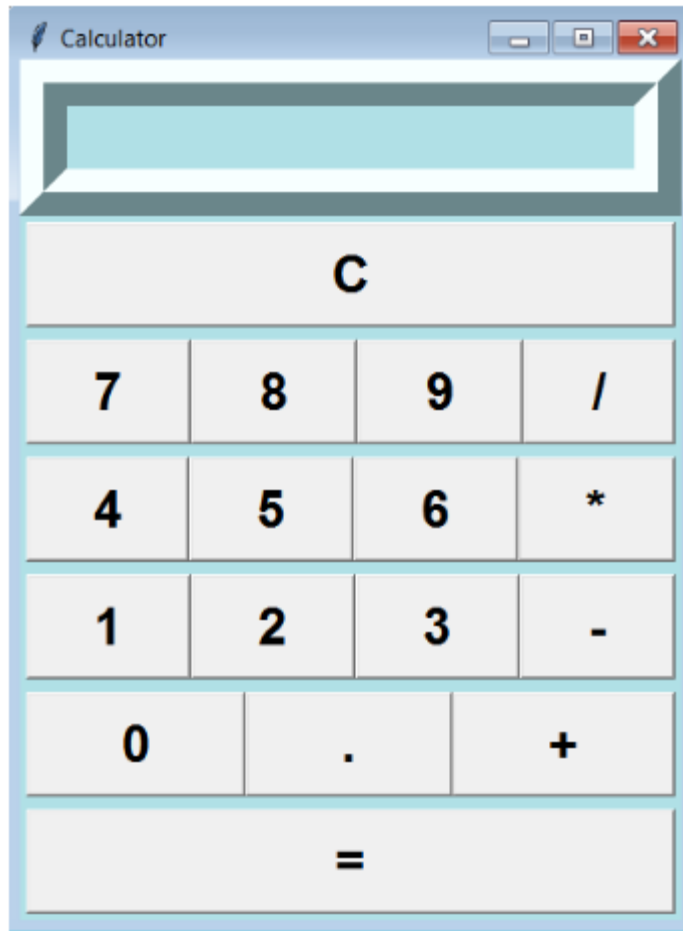


| Laboratory Activity No. 11 | |
|--|--|
| The Grid Manager | |
| Course Code: CPE103 | Program: BSCPE |
| Course Title: Object-Oriented Programming | Date Performed: 05-04-25 |
| Section: 1-A | Date Submitted: 05-04-25 |
| Name: Ruperto, April Anne A. | Instructor: Engr. Maria Rizette H. Sayo |
| 1. 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: | |
| <p>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.</p> <p>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</p> <p>Grids</p> <ul style="list-style-type: none"> 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: | |
| <p>Desktop Computer with Pycharm</p> <p>Windows Operating System</p> | |
| 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.

Please refer to this link: [CPE-103-OOP-1A/PyCharm Projects/Laboratory 11/Laboratory_11.py at main · Ruperto-April-Anne/CPE-103-OOP-1A](#)

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 this link: [CPE-103-OOP-1A/PyCharm Projects/Laboratory 11/Supplementary_Act_L11.py at main · Ruperto-April-Anne/CPE-103-OOP-1A](https://github.com/RupertoAprilAnne/CPE-103-OOP-1A-PyCharm-Projects-Laboratory-11-Supplementary_Act_L11.py)

Questions

1. How do you configure rows and columns in PyCharm when using Tkinter's grid() manager?
 - To set the configuration of rows and columns of the parent widget with the grid(), you use the grid_rowconfigure() and the grid_columnconfigure() methods. These methods set the weight of each row and column that determines how much space they use when the window is resized. Therefore, you give a weight value (which normally would be 1) to make grid resize proportionally, thus allowing the layout to adjust.
2. Why do widgets sometimes disappear when using grid() in PyCharm, and how can you fix it?
 - If you are combining geometry managers, for example grid() and pack(), in the same parent container, Widgets may disappear. Tkinter forbids this on the same widget, resulting in layout problems. Or it might be that are some badly positioned row/column widgets, or those widgets are placed outside visible area. The solution for this is to use only one geometry manager and correctly adjust the rows and columns to the grid with the grid_rowconfigure() and grid_columnconfigure().
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?
 - They tend to be used when a user action needs confirmation, an error occurred, etc. They allow an application to be more interactive and intuitive. If using grid() to align widgets across multiple frames, then you have to use the same row and column indices for the same widgets across the different frames. Similarly, since you are drawing all the widgets on a uniform grid layout, use columnspan or rowspan to make widgets span across multiple columns or rows, thus maintaining proper alignment across frames.

7. Conclusion:

- In conclusion, creating responsive rows and columns with grid () you must set the weight of each row and column. It only means that we cannot use grid(axis)/pack() in the same parent container at the same time or it gives error, like disappearing widgets. Message boxes collect up user input and allow them to interact, which makes better the user experience. The same effect can be achieved with grid() by making sure that the row and column indices for each frame are consistent, and using columnspan or rowspan to maintain alignment across frames for a neat, uniform pattern.

8. Assessment Rubric: