|  |  |
| --- | --- |
| **Laboratory Activity No. 11** | |
| **The Grid Manager** | |
| **Course Code:** CPE103 | **Program:** BSCPE |
| **Course Title:** Object-Oriented Programming | **Date Performed:** April 05, 2025 |
| **Section:** 1A | **Date Submitted:** April 05, 2025 |
| **Name:** Maringal, Czer Justine D. | **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:   * 1. Identify the main components in a GUI Application   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:  Calendar  Description automatically generated |

|  |
| --- |
| 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 Tkinter’s grid() we can use grid\_columnconfigure() and grid\_rowconfigure()   1. Why do widgets sometimes disappear when using grid() in PyCharm, and how can you fix it?   I think because of the incorrect row or column placements, or forgetting the root.mainloop()  It can be fixed by ensuring correct row-column setup and avoid conflict geometry managers   1. 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 provide alerts, feedback, and confirmations, making GUI application more user-friendly. I aligned the widgets across multiple frames by properly structuring the frames within the main window, using grid\_rowconfigure and grid\_columnconfigure |
| **7. Conclusion:** |
| I've learned how to structure a Tkinter GUI using grid(), configure rows and columns for responsive design, and troubleshoot disappearing widgets. I now understand how message boxes improve user experience and how to align widgets across multiple frames efficiently. This knowledge helps me to create functional and user-friendly Python applications. |
| **8. Assessment Rubric:** |