|  |  |
| --- | --- |
| **Laboratory Activity No. 11** | |
| **The Grid Manager** | |
| **Course Code:** CPE103 | **Program:** BSCPE |
| **Course Title:** Object-Oriented Programming | **Date Performed:** 04/05/25 |
| **Section:** BSCpE 1A | **Date Submitted:** 04/05/25 |
| **Name:** Nerio, Hannah Grace A. | **Instructor:** Engr. Maria Rizzete 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:   * 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.    **PLEASE REFER TO LAB 11 THAT I POSTED IN GITHUB UPON CHECKING THIS** |
| **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 LAB 11 THAT I POSTED IN GITHUB UPON CHECKING THIS** |

|  |
| --- |
| **Questions**   1. How do you configure rows and columns in PyCharm when using Tkinter's grid() manager?   When using Tkinter’s grid() manager, you can configure rows and columns by defining their weight and size using .grid\_rowconfigure() and .grid\_columnconfigure(). This helps ensure widgets expand proportionally when resizing the window.   1. Why do widgets sometimes disappear when using grid() in PyCharm, and how can you fix it?   Widgets may disappear when using grid() due to missing or incorrect placements within the layout. This can happen if the parent container (such as Tk() or Frame) does not have a defined size or is not packed properly. To resolve this issue, ensure that each widget is correctly assigned to a valid row and column using .grid(row=x, column=y). Also, check if the parent widget is properly configured and visible—sometimes, a forgotten .pack() or .grid() call for the parent frame might cause issues.     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 improve user experience by providing alerts, confirmations, and feedback, guiding users through actions while reducing errors. Tkinter's messagebox functions (showinfo(), showwarning(), showerror()) help make interactions smoother.  For aligning widgets across multiple frames with grid(), maintain consistency in row and column configurations. Use .grid\_rowconfigure() and .grid\_columnconfigure() to ensure proportional resizing, and apply sticky='nsew' to keep widgets properly positioned, creating a structured and visually balanced layout. |
| **7. Conclusion:** |
| In this activity, I learned how to use the grid () geometry manager in Tkinter to design and organize a GUI layout. I was able to apply it by creating a calculator interface and experimenting with different functions like arithmetic operations and even trigonometric ones. This activity helped me better understand how widgets can be placed and aligned properly using rows and columns. It also showed me the importance of user interface design and how message boxes and clear layouts can improve user experience. |
| **8. Assessment Rubric:** |