|  |  |
| --- | --- |
| **Laboratory Activity No. 11** | |
| **The Grid Manager** | |
| **Course Code:** CPE103 | **Program:** BSCPE |
| **Course Title:** Object-Oriented Programming | **Date Performed: 04/17/25** |
| **Section: 1A** | **Date Submitted: 04/17/25** |
| **Name: Eulin, Ryan Bertrand B.** | **Instructor: Ma’am 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?   It configure using grid() manager by using row, clumn, rowspa, and columnspan and sticky   1. Why do widgets sometimes disappear when using grid() in PyCharm, and how can you fix it?   Widgets disappear if the parent container’s size doesn’t expand to fit them which  Can be fixed using grid() properly and calling   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 instant feedback or alerts  While aligning widgets across multiple frames with grid() requires consistent use of  Sticky, padx, pady for layout |
| **7. Conclusion:** |
| Tkinter's grid() manager offers a powerful and flexible way to organize widgets in rows and columns, making GUI design more structured. Proper configuration of layout options like row, column, sticky, padx, and pady is essential for achieving clean alignment and visibility. Disappearing widgets are often due to layout mismanagement, which can be resolved with careful use of geometry management methods. Additionally, integrating message boxes helps make applications more interactive and user-friendly by providing clear feedback and guidance. |
| **8. Assessment Rubric:** |