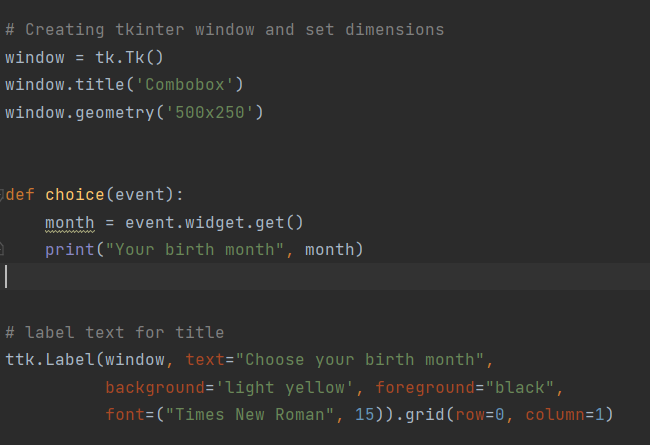
|  |  |
| --- | --- |
| **Laboratory Activity No. 10** | |
| **The Selection Widgets using Pycharm** | |
| **Course Code:** CPE103 | **Program:** BSCPE |
| **Course Title:** Object-Oriented Programming | **Date Performed:** 03/22/25 |
| **Section:** BSCpE 1A | **Date Submitted:** 03/22/25 |
| **Name:** Nerio, Hannah Grace A. | **Instructor:** Maria Rizzete H. Sayo |
| **1. Objective(s):** | |
| This activity aims to familiarize students with the Pycharm framework and selection widget | |
| **2. Intended Learning Outcomes (ILOs):** | |
| The students should be able to:   * 1. To create a Python program that use selection widget like Combobox   2. To use ttk function as part of Tk ( ) in the Tkinter module | |
| **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.  Pycharm is an integrated development environment used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django. | |
| **4. Materials and Equipment:** | |
| Desktop Computer with Anaconda Python or Pycharm Windows Operating System | |
| **5. Procedure:** | |



|  |
| --- |
| 1.   1. Run the program and observe the output.   **Adding an icon**   1. Download any .ico picture from <https://icon-icons.com/> or any similar sites. 2. Place the icon in your folder (ex. Oopfa1<lastname>\_lab10)     5. Run the program again, the program should now have an icon similar to the program below. |
| **6. Supplementary Activity:** |
| **Task**  1. Create label widgets below to label your birth date <dd>, birth year <yyyy>  2. Create combobox to drop down your birth date <dd>, birth year <yyyy>  3. Create another method to show info about your birth date <dd>, birth year <yyyy>  Note: You may also use additional selection(listbox, radio button, check button) or common widgets to improve the design of your GUI. |
| **Questions**   1. What are selection widgets?   A selection widgets are interactive tools that allow users to make choices or input data, typically within a graphical user interface (GUI). These widgets can be created using libraries like Tkinter, PyQt, or Kivy. For example, drop-down menus let users select an option from a list, checkboxes allow multiple choices, radio buttons enable selecting only one option, and sliders let users adjust values. These widgets make Python programs more user-friendly by enabling easy interaction.   1. Which Python libraries provide selection widgets?   Several Python libraries provide selection widgets to create interactive graphical user interfaces (GUIs). **Tkinter**, a built-in Python library, offers basic widgets like drop-down menus, checkboxes, radio buttons, and sliders for simple GUI applications. **PyQt** and **PySide** are more advanced libraries that provide highly customizable widgets and are often used for complex applications. **Kivy** is another popular library designed for modern, touch-friendly GUIs, supporting selection widgets like spinners and sliders. Additionally, **Dear PyGui** offers a GPU-powered framework for creating responsive widgets, and **WxPython** is known for providing native-looking GUI components, including selection widgets.   1. How do selection widgets enhance user interaction in GUI applications?   Selection widgets enhance user interaction in GUI applications by making it easier for users to input data and navigate options. They provide intuitive and visually organized ways to make choices, such as selecting from drop-down menus, ticking checkboxes, or adjusting values with sliders. These widgets simplify complex tasks, reduce user error, and improve accessibility by offering clear, interactive controls. By catering to a range of user preferences, selection widgets create smoother and more engaging experiences in applications. |
| **7. Conclusion:** |
| In this laboratory activity, we explored selection widgets in Python using Tkinter and PyCharm. We learned how to create and use widgets like Combobox, checkboxes, and radio buttons to enhance user interaction in GUI applications. These tools make programs more user-friendly by allowing users to select options easily. Understanding selection widgets helps in developing better interfaces, improving accessibility, and making applications more interactive. Through hands-on practice, we gained practical experience in building GUI components, which is essential for creating effective and engaging software. |
| **8. Assessment Rubric:** |