

Laboratory Activity No. 10	
The Selection Widgets using Pycharm	
Course Code: CPE103	Program: BSCPE
Course Title: Object-Oriented Programming	Date Performed: March 29, 2025
Section: 1A	Date Submitted: March 30, 2025
Name: Czer Justine D. Maringal	Instructor: Engr. Maria Rizette 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: 2.1 To create a Python program that use selection widget like Combobox 2.2 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:	

```
# Creating tkinter window and set dimensions
window = tk.Tk()
window.title('Combobox')
window.geometry('500x250')

def choice(event):
    month = event.widget.get()
    print("Your birth month", month)

# label text for title
ttk.Label(window, text="Choose your birth month",
          background='light yellow', foreground="black",
          font=("Times New Roman", 15)).grid(row=0, column=1)
```

1.

```
month.grid(column=1, row=5)
month.current()

def choice(event):
    showinfo(
        title = "Selection",
        message = f'You selected {n.get()}')

month.bind("<<ComboboxSelected>>", choice)
window.mainloop()
```

2. Run the program and observe the output.

Adding an icon

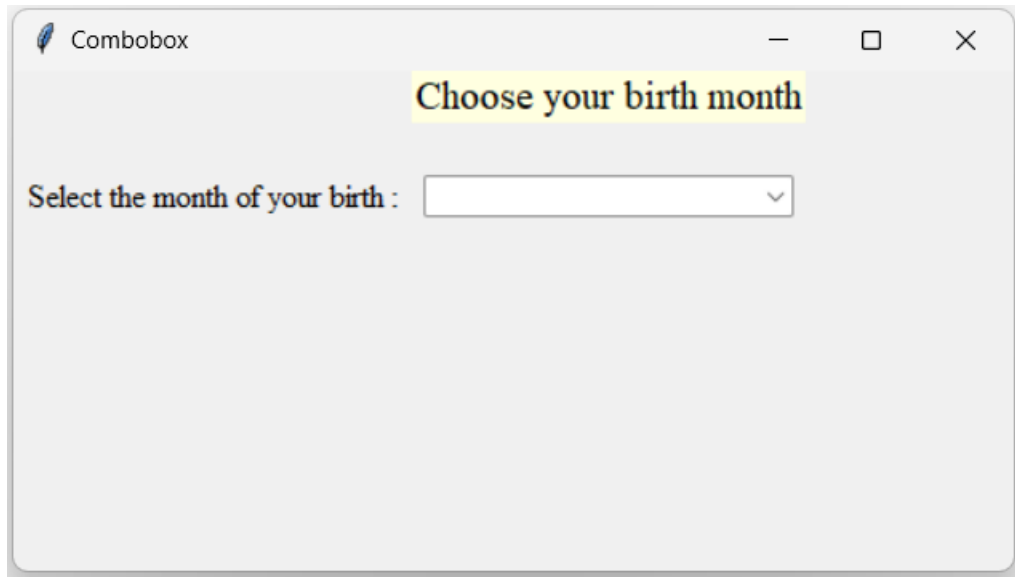
3. Download any .ico picture from <https://icon-icons.com/> or any similar sites.
4. Place the icon in your folder (ex. Oopfa1<lastname>_lab10)

```
# Set label
ttk.Label(window, text="Select the month of your birth :",
          font=("Times New Roman", 12)).grid(column=0,
          row=5, padx=5, pady=25)

# Create Combobox
n = tk.StringVar()
month = ttk.Combobox(window, width=27, textvariable=n)

# Adding combobox drop down list
month['values'] = (' January',
                  ' February',
                  ' March',
                  ' April',
                  ' May',
                  ' June',
                  ' July',
                  ' August',
                  ' September',
                  ' October',
                  ' November',
                  ' December')
```

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?

Selection widgets are interactive elements in a graphical user interface (GUI) that allow users to choose from multiple options. Common types include radio buttons, which enable selecting only one option from a group, check buttons, which allow multiple selections, list boxes, which display a list of selectable items, and menus, which provide dropdown selections

2. Which Python libraries provide selection widgets?

Python offers several libraries for creating selection widgets in graphical user interfaces (GUIs). **Tkinter** is a built-in library that provides radio buttons, check buttons, and dropdown menus for basic GUI applications. **PyQt** and **PySide** offer more advanced widgets for complex desktop applications.

3. How do selection widgets enhance user interaction in GUI applications?

Selection widgets simplify user input by offering clear, organized choices, reducing errors, saving space, and speeding up the process. They improve efficiency and the overall user experience.

- 4.

7. Conclusion:

Widgets play a crucial role in creating interactive and user-friendly applications. They enhance usability by providing intuitive ways for users to interact with software, whether through selection widgets like checkboxes, radio buttons, and dropdowns or input fields and sliders. By streamlining user choices and navigation, widgets improve efficiency, accessibility, and overall user experience.

8. Assessment Rubric: