Laboratory Activity No. 10	
The Selection Widgets using Pycharm	
Course Code: CPE103	**Program:** BSCPE
Course Title: Object-Oriented Programming	**Date Performed:** 03/22/2025
Section: 1-A	**Date Submitted:** 03/22/2025
Name: BRON, JHUSTINE A.	**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

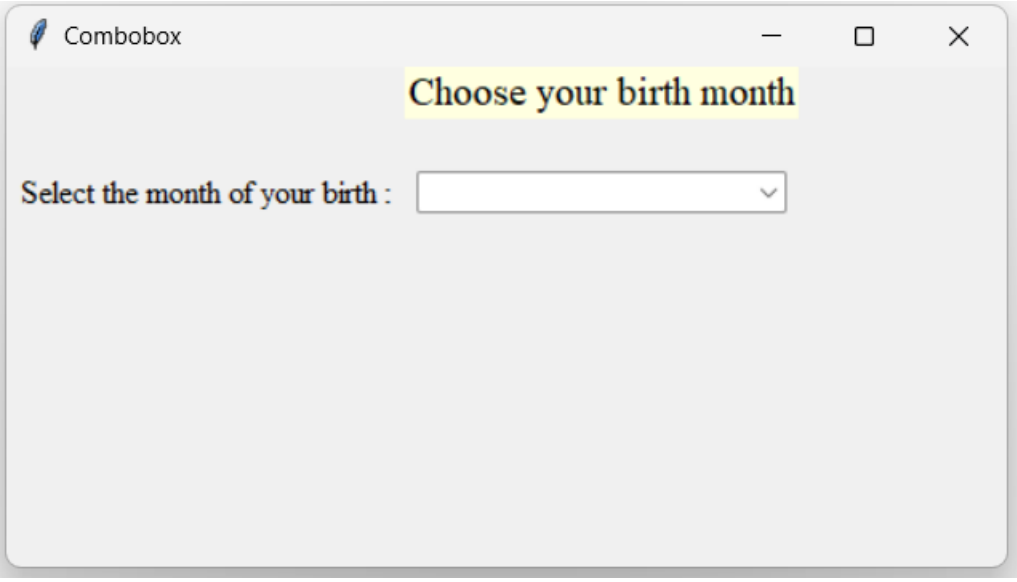
3. Download any .ico picture from https://icon-icons.com/ or any similar sites.
4. Place the icon in your folder (ex. Oopfa1<lastname>_lab10)

```
# 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 tools in GUIs like drop-down menus, radio buttons, checkboxes, and sliders. They let users easily choose from a set of options, making interactions more seamless.

2. Which Python libraries provide selection widgets?

   - Popular Python libraries offering these tools include **Tkinter**, **PyQt/PySide**, **Kivy**, **Dash**, and **Streamlit**. Each provides various widgets to create interactive apps.

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

   - Selection widgets make apps more intuitive and user-friendly. They reduce typing errors, present options clearly, and create engaging, efficient, and visually appealing interfaces.

7. Conclusion:

- Selection widgets are essential components of graphical user interfaces (GUIs) that improve the overall user experience by allowing efficient and intuitive interactions. Python libraries such as Tkinter, PyQt/PySide, Kivy, Dash, and Streamlit empower developers to create applications that are not only functional but also visually appealing and accessible. By integrating selection widgets like drop-down menus, checkboxes, sliders, and radio buttons, developers can minimize user errors, clearly present choices, and create engaging interfaces that streamline user input and interactions.

8. Assessment Rubric: