

ACTIVITY NO. 8

Laboratory Activity 8 - The Selection Widget

Course Code: CPE009B	Program: Computer Engineering
Course Title: Object Oriented Programming	Date Performed: 11/25/24
Section: CPE 21S4	Date Submitted: 11/25/24
Group Members: Virtucio, Dominic Joseph Bonifacio, Nyko Adrein Magistrado, Aira Pauleen Planta, Calvin Earl Solis, Paul Vincent	Instructor: Ma'am Maria Rizette Sayo

III. Results

test.py code

```
Python
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
from tkinter.messagebox import showinfo

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

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

# 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)

# 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)

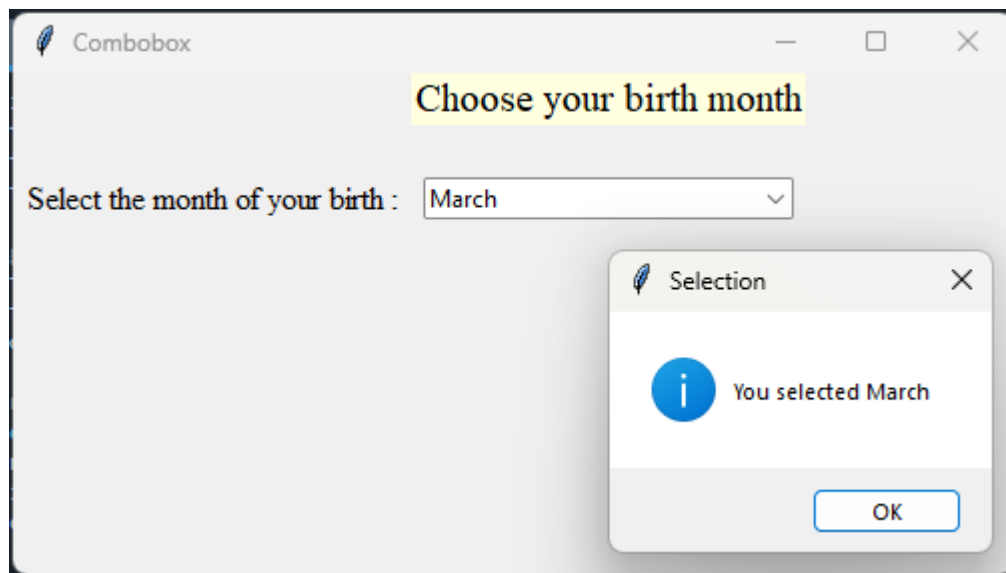
# Create Combobox
n = tk.StringVar()
month = ttk.Combobox(window, width=27, textvariable=n)

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

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

# Binding the selection event
month.bind("<<ComboboxSelected>>", choice)

# Run the tkinter main loop
window.mainloop()
```



Output of test.py

Insights

One of the most important things We learned from this program is using event-driven programming. It allows the program to dynamically respond to what We do at that time, like selecting a month from the combobox. To this end, by binding the `<<ComboboxSelected>>` event to the `choice()` function, We can create specific actions, like showing the message box, depending on the selected option. These actions can be seen in the code by requiring the user to choose which month they would like, and then displaying a message confirming their choice. This is very useful because it would show whether they were able to choose the correct option.

We also understood how useful data binding really is using tkinter. We don't have to manually track what is selected using `StringVar()` to bind the combobox to a variable. We can simply retrieve the selected month using `n.get()`, keeping the code way cleaner and easier to maintain.

We are also impressed by the themed widgets of the `ttk` module. We can easily make my application look something like modern with these and avoid those older tkinter widgets.

	<p>They make the interface look more professional and friendly to use.</p> <p>The message box is a great idea; because it lets me know right away what We just chose. It is an easy way of confirming my action and ensures that we know what we just chose.</p>
Commentaries	<p>This program is a good beginning point in how to construct interactive applications with tkinter. It is very helpful since it demonstrates the basic concepts needed for creating a functional GUWe application. The labels guide me step by step through the process, and the combobox lets me choose a month without confusion. Then, a message box pops up to confirm my choice, which shows how an interactive program can be very helpful in many things like user's satisfaction.</p> <p>The code is an event-driven approach. It's not a static interface presentation; rather, it responds to my actions. We click a month, and it shows a clarification message to ensure that I'm selecting the correct option. The program gives instant feedback, which is the essence of computer programming as this would allow the users to deal with an interface that can respond immediately</p> <p>The grid layout manager also makes it easy to structure everything neatly. That is, it makes it easy for me to specify exactly where We want each widget, and it gives me a structured interface. The overall look and feel are simple yet effective. Everything is logically laid out, and the spacing ensures the interface isn't too cluttered.</p>
Explanation	<p>This is a simple Python program that gives a GUWe from tkinter. It creates a window where it gives an option of the user's birth month by a choice in the dropdown list (combobox). After selection, a pop-up information shows which month was selected. The program begins with importing the modules necessary for the program, such as tkinter, ttk, and messagebox. It then creates a main window and positions the widgets in the grid layout manager. There are placed labels to guide the user and a combobox filled up with the months of the year. As soon as the user selects a month, it automatically calls the choice() function that will prompt for a message box showing the month he picked.</p>

IV. Conclusion

This Python program demonstrates how to create a simple graphical user interface (GUI) using Tkinter. It features a dropdown menu (combobox) that allows users to select their birth month. After making a selection, a message box appears displaying the chosen month. Through this program, we see the basic concepts we've learned in oop. It shows how to create a window with various widgets, like labels and comboboxes, and how to handle events triggered by user interactions. When the user selects a month, the program responds by showing feedback in the form of a message box. This example highlights how we can build interactive, user-friendly applications with Tkinter. We aslo gained a comprehensive understanding of event-driven programming, where the program reacts dynamically to user interactions, such as selecting a month from the combobox. We learned how to bind events to functions, enabling specific actions based on user choices. The use of data binding

with StringVar() simplified the process of retrieving selected values, making our code cleaner and more maintainable. We also explored the advantages of themed widgets from the ttk module, enhancing the visual appeal and user experience of our application. Through this project, we mastered the creation of interactive user interfaces, incorporating elements like drop-down menus, labels, and message boxes. We successfully implemented a GUI that provides feedback to the user, confirming their selection and ensuring a seamless user experience.

V. Reference

https://www.tutorialspoint.com/python/python_gui_programming.htm

<https://stackoverflow.com/questions/66552988/python-tkinter-run-python-script-when-click-button-widget-of-tkinter-and-keep-v>