Module 4: Python GUI Development with Tkinter

4.1 Introduction to Tkinter

4.1.1 What is Tkinter?

- **Tkinter** is Python's standard GUI (Graphical User Interface) library.
- It provides an easy-to-use interface for creating desktop applications.
- It is built on **Tk**, a robust cross-platform windowing toolkit.

Why Tkinter?

- **Built-in Library:** Comes pre-installed with Python.
- **Lightweight:** Ideal for small-to-medium-scale GUI applications.
- **Cross-Platform:** Works on Windows, macOS, and Linux.

4.1.2 Installing Tkinter

Tkinter is included with standard Python installations. To verify:

```
import tkinter as tk
print(tk.TkVersion)
```

If it's not available, install it via your package manager:

```
sudo apt-get install python3-tk  # For Linux
```

4.2 Creating a Basic Tkinter Window

```
import tkinter as tk

# Create main window
root = tk.Tk()

# Set window title and size
root.title("My First Tkinter App")
root.geometry("400x300")

# Start the Tkinter event loop
root.mainloop()
```

4.3 Tkinter Widgets

Widget	Description	Example
Label	Displays text or images	`tk.Label()`
Button	Triggers an action	`tk.Button()`
Entry	Single-line text input	`tk.Entry()`
Text	Multi-line text input	`tk.Text()`
Frame	Container for other widgets	`tk.Frame()`
Checkbutton	Checkbox input	`tk.Checkbutton()`
Radiobutton	Select one option from many	`tk.Radiobutton()`
Listbox	List of selectable options	`tk.Listbox()`
Combobox	Dropdown selection	`ttk.Combobox()`

```
import tkinter as tk
from tkinter import ttk

# Main Tkinter window
window = tk.Tk()
window.title("Tkinter Widgets Example")
window.geometry("500x500")

# Label widget
label = tk.Label(window, text="This is a label")
label.pack(pady=10)

# Button widget
button = tk.Button(window, text="Click Me")
button.pack(pady=10)

# Entry widget
entry = tk.Entry(window)
entry.pack(pady=10)

# Text widget
text = tk.Text(window, height=5, width=30)
text.pack(pady=10)

# Checkbutton widget
check_var = tk.IntVar()
checkbutton = tk.Checkbutton(window, text="Check me", variable=check_var)
checkbutton.pack(pady=10)

# Radiobutton widget
radio_var = tk.StringVar(value="Option1")
radiobutton1 = tk.Radiobutton(window, text="Option 1", variable=radio_var,
value="Option1")
radiobutton2 = tk.Radiobutton(window, text="Option 2", variable=radio_var,
value="Option2")
radiobutton1.pack(pady=5)
radiobutton2.pack(pady=5)
```

```
# Listbox widget
listbox = tk.Listbox(window)
listbox.insert(1, "Item 1")
listbox.insert(2, "Item 2")
listbox.insert(3, "Item 3")
listbox.pack(pady=10)

# Frame widget
frame = tk.Frame(window, bg="lightgray", width=100, height=50)
frame.pack(pady=10)

# Scale widget
scale = tk.Scale(window, from_=0, to=100, orient="horizontal")
scale.pack(pady=10)

# Spinbox widget
spinbox = tk.Spinbox(window, from_=1, to=10)
spinbox.pack(pady=10)

# Combobox widget
combobox = ttk.Combobox(window, values=["Option 1", "Option 2", "Option 3"])
combobox.pack(pady=10)
combobox.place(x=50,y=60)

# Menu widget

menu = tk.Menu(window)

window.config(menu=menu)
file_menu = tk.Menu(menu)
menu.add_cascade(label="File", menu=file_menu)
file_menu.add_command(label="Open")
file_menu.add_command(label="Save")
file_menu.add_separator()
file_menu.add_command(label="Exit", command=window.quit)

window.mainloop()
```

4.4 Widget Placement Methods

Example:

```
import tkinter as tk

# Create main window
root = tk.Tk()

# Set window title and size
root.title("My First Tkinter App")
root.geometry("400x300")


label = tk.Label(root, text="Hello, Tkinter!")
label.pack()  # Uses pack layout, this will
root.mainloop()


# Start the Tkinter event loop
root.mainloop()
```

4.5 Event Handling in Tkinter

4.5.1 Handling Button Clicks

```
import tkinter as tk

def on_click():
    print("Button clicked!")

root = tk.Tk()
root.geometry("300x300")
button = tk.Button(root, text="Click Me", command=on_click)
button.pack()
root.mainloop()
```

4.5.2 Binding Events

Common Events:

Event	Description
<Button-1>	Left mouse click
<Button-3>	Right mouse click
<Return>	Enter key press
<KeyPress>	Any key press
<Double-1>	mouse left double click
<Enter>	mouse enters the window
<Leave>	mouse leaves the window
<Motion>	tracks the motion of mouse
<FocusIn>	widget gets focus
<FocusOut>	widget losts focus
<MouseWheel>	mouse wheel event

```python
import tkinter as tk

# Function to handle different events
def button_click(event):
    print("Button clicked!")

def button_double_click(event):
    print("Button double-clicked!")

def button_right_click(event):
    print("Button right-clicked!")

def key_press(event):
    print(f"You pressed '{event.char}'")

def enter_widget(event):
    print("Mouse entered the widget")

def leave_widget(event):
    print("Mouse left the widget")

def motion_event(event):
    print(f"Mouse moving: ({event.x}, {event.y})")

def focus_in(event):
    print("Widget got focus")

def focus_out(event):
    print("Widget lost focus")

def mouse_wheel(event):
    print("Mouse wheel used")

#-----------------------------------------------------------------------------
window = tk.Tk()
window.title("Tkinter Binding Events Example")
window.geometry("400x300")
```

```
button = tk.Button(window, text="Click Me")
button.pack(pady=20)

entry = tk.Entry(window)
entry.pack(pady=20)
#-------------------------------------------------------------------------------
# Binding events
button.bind("<Button-1>", button_click)           # Left click
button.bind("<Double-1>", button_double_click)    # Double left click
button.bind("<Button-3>", button_right_click)     # Right click
window.bind("<Key>", key_press)                   # Any key press
button.bind("<Enter>", enter_widget)              # Mouse enters widget
button.bind("<Leave>", leave_widget)              # Mouse leaves widget
button.bind("<Motion>", motion_event)             # Mouse movement over widget
entry.bind("<FocusIn>", focus_in)                 # Widget gets focus
entry.bind("<FocusOut>", focus_out)               # Widget loses focus
window.bind("<MouseWheel>", mouse_wheel)          # Mouse wheel event

window.mainloop()
```

4.6 Tkinter Layout Management

4.6.1 Using pack()

```
import tkinter as tk
root = tk.Tk()
tk.Label(root, text="Top Label").pack(side="top")
tk.Label(root, text="Bottom Label").pack(side="bottom")
root.mainloop()
```

4.6.2 Using grid()

```
import tkinter as tk


root = tk.Tk()
root.geometry("500x500")
tk.Label(root, text="Top Label").pack(side="top")
tk.Label(root, text="Bottom Label").pack(side="bottom")
root.mainloop()
```

4.6.3 Using place()

```
import tkinter as tk
root = tk.Tk()
tk.Label(root, text="Placed Label").place(x=100, y=50)
tk.Label(root, text="Placed Label").place(x=50, y=100)
root.mainloop()
```

4.7 Tkinter Messagebox

```
import tkinter as tk
from tkinter import messagebox

def show_message():
    messagebox.showinfo("Info", "This is a Tkinter messagebox")

root = tk.Tk()
root.geometry("300x300")
tk.Button(root, text="Show Message", command=show_message).pack()
root.mainloop()
```

4.8 Tkinter File Dialog

```
import tkinter as tk


from tkinter import filedialog

def open_file():
    file = filedialog.askopenfilename()
    print(f"File selected: {file}")

root = tk.Tk()
tk.Button(root, text="Open File", command=open_file).pack()
root.mainloop()
```

4.9 Tkinter Project: Simple Login Form

```
import tkinter as tk
from tkinter import messagebox

def login():
    username = entry_username.get()
    password = entry_password.get()
    if username == "admin" and password == "1234":
        messagebox.showinfo("Success", "Login Successful")
    else:
        messagebox.showerror("Error", "Invalid Credentials")

root = tk.Tk()
root.title("Login Form")
root.geometry("300x200")

# Labels
tk.Label(root, text="Username:").pack(pady=5)
entry_username = tk.Entry(root)
entry_username.pack(pady=5)

tk.Label(root, text="Password:").pack(pady=5)
entry_password = tk.Entry(root, show="*")
entry_password.pack(pady=5)

# Button
tk.Button(root, text="Login", command=login).pack(pady=10)

root.mainloop()
```

4.10 Advanced Tkinter Concepts

4.10.1 Frames and Nested Layouts

```
import tkinter as tk
root = tk.Tk()
root.geometry("300x300")
frame = tk.Frame(root)
```

```
frame.pack()

tk.Label(frame, text="Inside Frame").pack()
root.mainloop()
```

4.10.2 Canvas for Graphics

```
import tkinter as tk


root = tk.Tk()
root.geometry("300x300")


canvas = tk.Canvas(root, width=200, height=100)
canvas.pack()

canvas.create_line(0, 0, 200, 100)
canvas.create_rectangle(50, 25, 150, 75, fill="blue")
root.mainloop()
```

4.10.3 Menus and Submenus

```
import tkinter as tk


root = tk.Tk()
root.geometry("300x300")
menu = tk.Menu(root)

file_menu = tk.Menu(menu, tearoff=0)
file_menu.add_command(label="Open")
file_menu.add_command(label="Save")
file_menu.add_separator()
file_menu.add_command(label="Exit", command=root.quit)

menu.add_cascade(label="File", menu=file_menu)
root.config(menu=menu)
root.mainloop()
```

4.11 Summary of Module 4 (Concepts we have Learned)