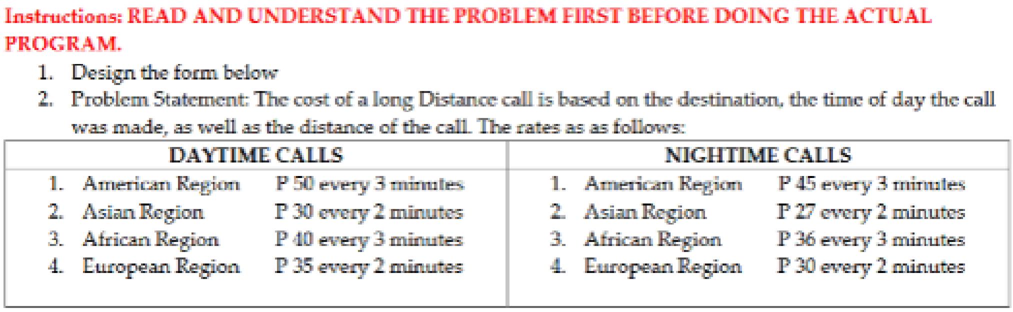
Bonillo, Loren
C204

Finals Lab Task 4. Python GUI using TKINTER

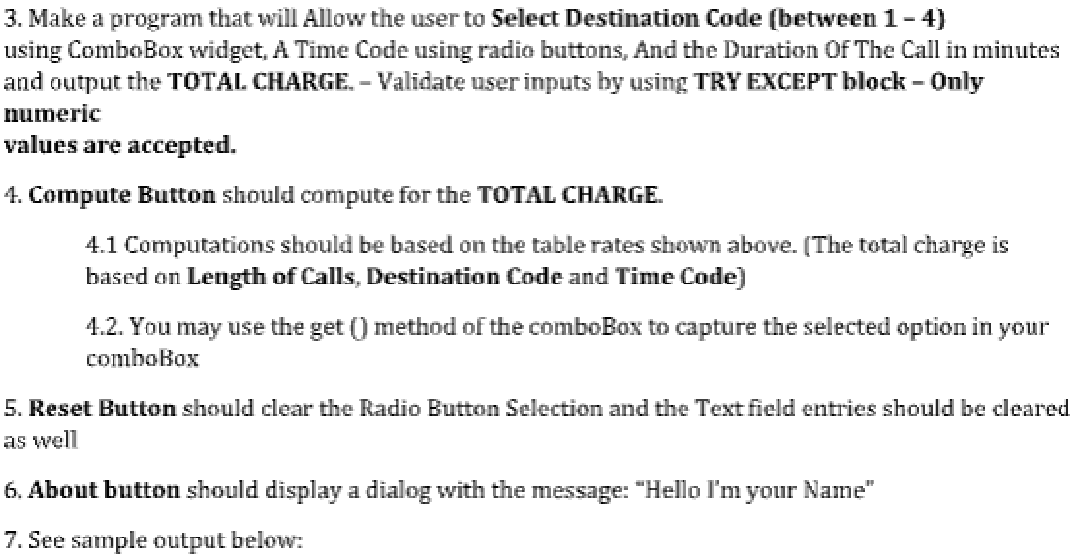
Note: Write your code following OOP code construct, you may use the attached simpleCalc.py program as guide.



Instructions: READ AND UNDERSTAND THE PROBLEM FIRST BEFORE DOING THE ACTUAL PROGRAM.

1. Design the form below
2. Problem Statement: The cost of a long Distance call is based on the destination, the time of day the call was made, as well as the distance of the call. The rates as as follows:

DAYTIME CALLS		NIGHTIME CALLS	
1. American Region	P 50 every 3 minutes	1. American Region	P 45 every 3 minutes
2. Asian Region	P 30 every 2 minutes	2. Asian Region	P 27 every 2 minutes
3. African Region	P 40 every 3 minutes	3. African Region	P 36 every 3 minutes
4. European Region	P 35 every 2 minutes	4. European Region	P 30 every 2 minutes



3. Make a program that will Allow the user to **Select Destination Code (between 1 – 4)** using ComboBox widget, A Time Code using radio buttons, And the Duration Of The Call in minutes and output the **TOTAL CHARGE.** – Validate user inputs by using **TRY EXCEPT block – Only numeric values are accepted.**

4. **Compute Button** should compute for the **TOTAL CHARGE.**

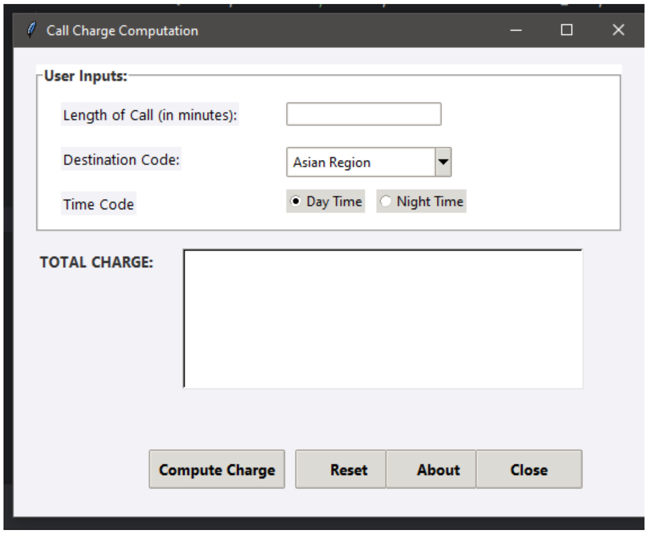
4.1 Computations should be based on the table rates shown above. (The total charge is based on **Length of Calls**, **Destination Code** and **Time Code**)

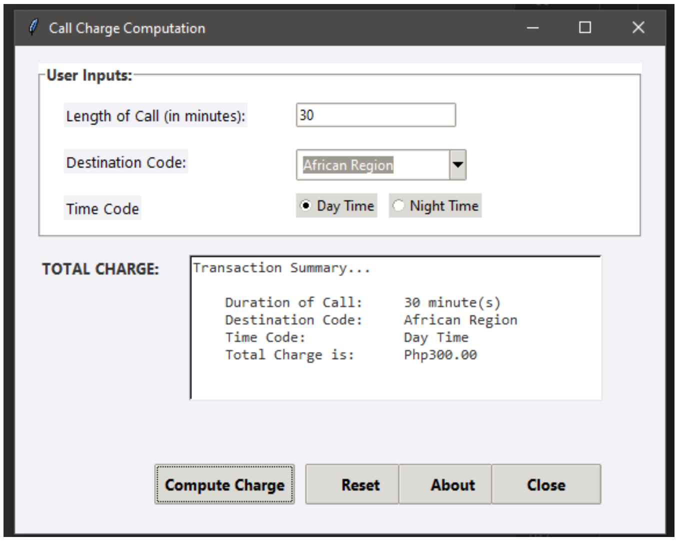
4.2. You may use the get () method of the comboBox to capture the selected option in your comboBox

5. **Reset Button** should clear the Radio Button Selection and the Text field entries should be cleared as well

6. **About button** should display a dialog with the message: "Hello I'm your Name"

7. See sample output below:

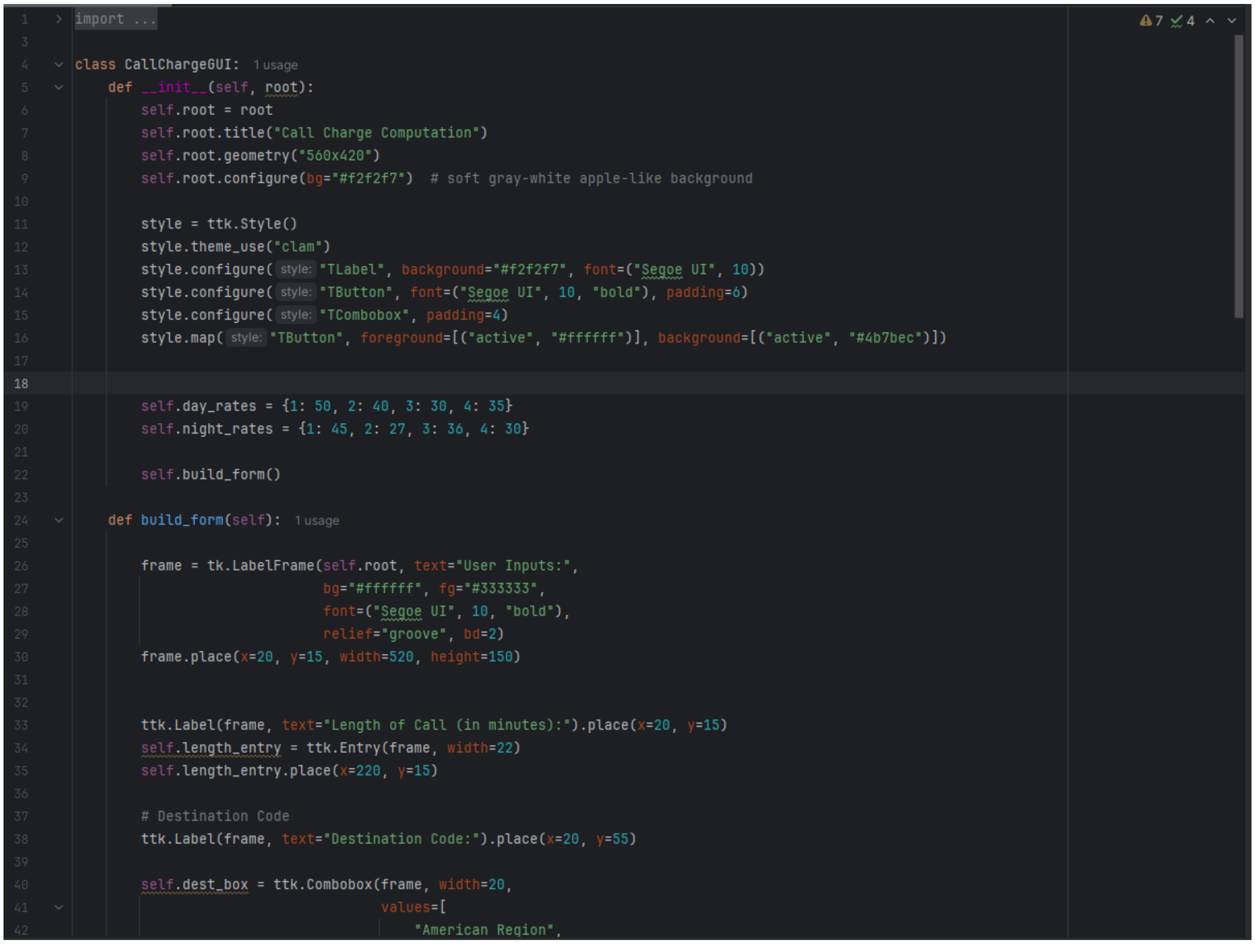




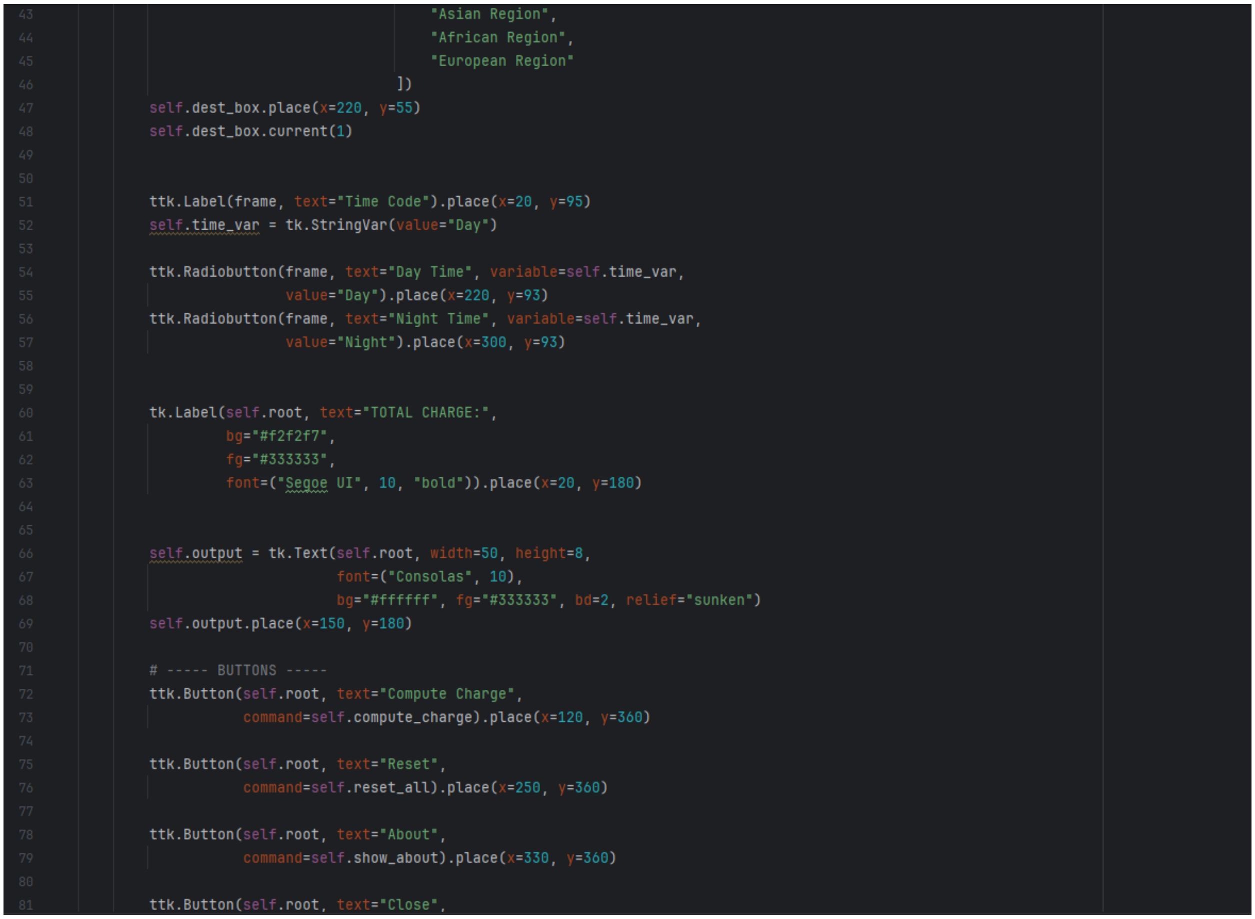
Length of Call (in minutes): 30
Destination Code: African Region
Time Code: ● Day Time ○ Night Time

```
Transaction Summary...

    Duration of Call:      30 minute(s)
    Destination Code:      African Region
    Time Code:             Day Time
    Total Charge is:       Php300.00
```

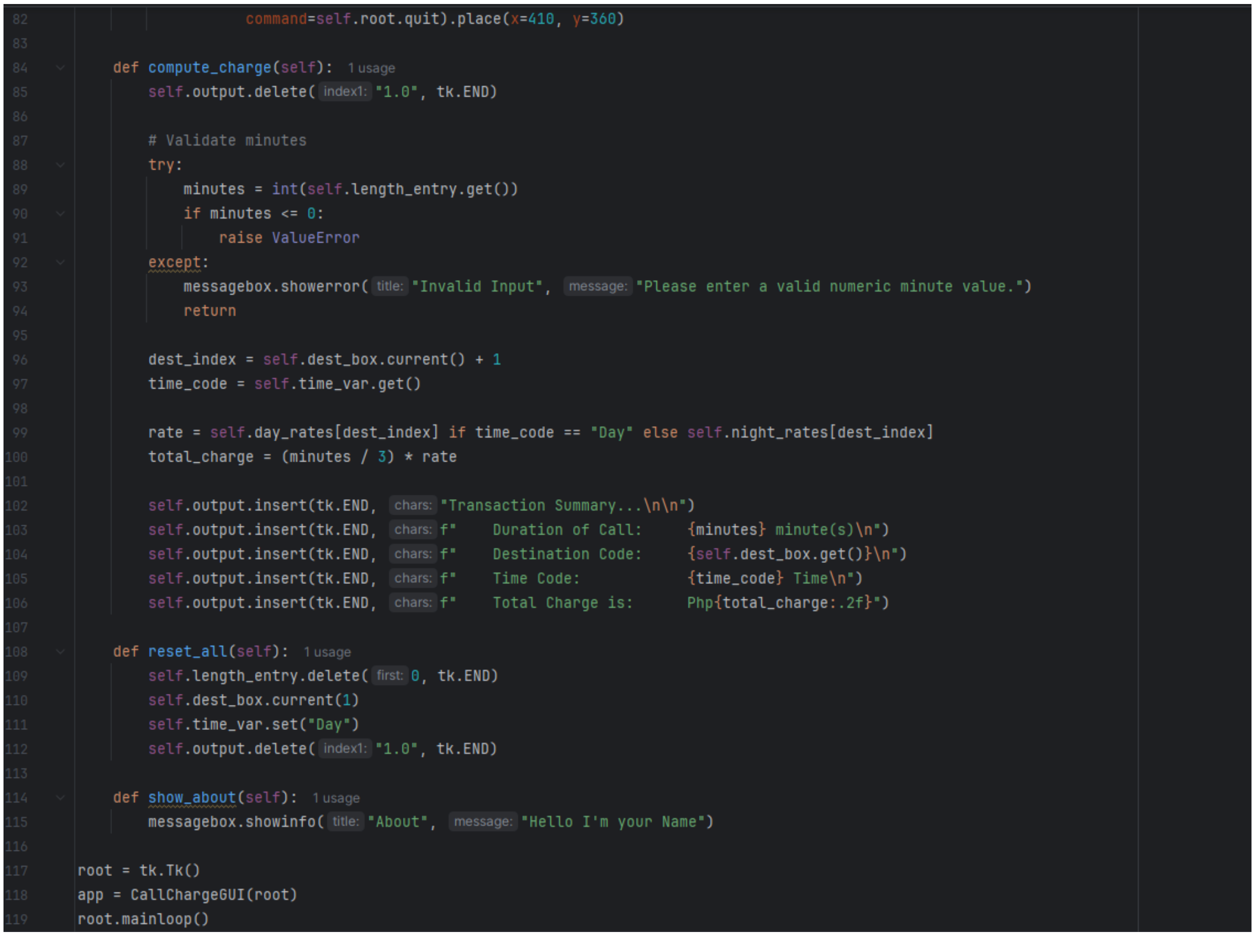
Source Code:



```python
import ...

class CallChargeGUI:
    def __init__(self, root):
        self.root = root
        self.root.title("Call Charge Computation")
        self.root.geometry("560x420")
        self.root.configure(bg="#f2f2f7")  # soft gray-white apple-like background

        style = ttk.Style()
        style.theme_use("clam")
        style.configure("TLabel", background="#f2f2f7", font=("Segoe UI", 10))
        style.configure("TButton", font=("Segoe UI", 10, "bold"), padding=6)
        style.configure("TCombobox", padding=4)
        style.map("TButton", foreground=[("active", "#ffffff")], background=[("active", "#4b7bec")])


        self.day_rates = {1: 50, 2: 40, 3: 30, 4: 35}
        self.night_rates = {1: 45, 2: 27, 3: 36, 4: 30}

        self.build_form()

    def build_form(self):

        frame = tk.LabelFrame(self.root, text="User Inputs:",
                              bg="#ffffff", fg="#333333",
                              font=("Segoe UI", 10, "bold"),
                              relief="groove", bd=2)
        frame.place(x=20, y=15, width=520, height=150)


        ttk.Label(frame, text="Length of Call (in minutes):").place(x=20, y=15)
        self.length_entry = ttk.Entry(frame, width=22)
        self.length_entry.place(x=220, y=15)

        # Destination Code
        ttk.Label(frame, text="Destination Code:").place(x=20, y=55)

        self.dest_box = ttk.Combobox(frame, width=20,
                                     values=[
                                         "American Region",
```



```
                                         "Asian Region",
                                         "African Region",
                                         "European Region"
                                     ])
        self.dest_box.place(x=220, y=55)
        self.dest_box.current(1)


        ttk.Label(frame, text="Time Code").place(x=20, y=95)
        self.time_var = tk.StringVar(value="Day")

        ttk.Radiobutton(frame, text="Day Time", variable=self.time_var,
                        value="Day").place(x=220, y=93)
        ttk.Radiobutton(frame, text="Night Time", variable=self.time_var,
                        value="Night").place(x=300, y=93)


        tk.Label(self.root, text="TOTAL CHARGE:",
                 bg="#f2f2f7",
                 fg="#333333",
                 font=("Segoe UI", 10, "bold")).place(x=20, y=180)


        self.output = tk.Text(self.root, width=50, height=8,
                              font=("Consolas", 10),
                              bg="#ffffff", fg="#333333", bd=2, relief="sunken")
        self.output.place(x=150, y=180)

        # ----- BUTTONS -----
        ttk.Button(self.root, text="Compute Charge",
                   command=self.compute_charge).place(x=120, y=360)

        ttk.Button(self.root, text="Reset",
                   command=self.reset_all).place(x=250, y=360)

        ttk.Button(self.root, text="About",
                   command=self.show_about).place(x=330, y=360)

        ttk.Button(self.root, text="Close",
```



```
                   command=self.root.quit).place(x=410, y=360)

    def compute_charge(self):  1 usage
        self.output.delete( index1: "1.0", tk.END)

        # Validate minutes
        try:
            minutes = int(self.length_entry.get())
            if minutes <= 0:
                raise ValueError
        except:
            messagebox.showerror( title: "Invalid Input",  message: "Please enter a valid numeric minute value.")
            return

        dest_index = self.dest_box.current() + 1
        time_code = self.time_var.get()

        rate = self.day_rates[dest_index] if time_code == "Day" else self.night_rates[dest_index]
        total_charge = (minutes / 3) * rate

        self.output.insert(tk.END,  chars: "Transaction Summary...\n\n")
        self.output.insert(tk.END,  chars: f"    Duration of Call:     {minutes} minute(s)\n")
        self.output.insert(tk.END,  chars: f"    Destination Code:     {self.dest_box.get()}\n")
        self.output.insert(tk.END,  chars: f"    Time Code:            {time_code} Time\n")
        self.output.insert(tk.END,  chars: f"    Total Charge is:      Php{total_charge:.2f}")

    def reset_all(self):  1 usage
        self.length_entry.delete( first: 0, tk.END)
        self.dest_box.current(1)
        self.time_var.set("Day")
        self.output.delete( index1: "1.0", tk.END)

    def show_about(self):  1 usage
        messagebox.showinfo( title: "About",  message: "Hello I'm your Name")

root = tk.Tk()
app = CallChargeGUI(root)
root.mainloop()
```