TAYTING, JUSTINE JAY D.
BSCS – C204

OOP
Dec. 06, 2025

**Python and Tkinter GUI program**
**Final Task 4**

This task requires creating an OOP-based Python Tkinter GUI program that computes the total charge of a long-distance call using a ComboBox for selecting destination code (1–4), radio buttons for choosing time of day (Daytime or Nighttime), and a text field for entering call duration in minutes. The computation must follow the given rate table, with different charges for daytime and nighttime calls across American, Asian, African, and European regions. The program must validate numeric input using try–except, and the Compute button should display a transaction summary showing duration, destination, time code, and total charge. A Reset button must clear all inputs and selections, while an About button should show a dialog message saying “Hello I’m <your name>.” The interface should be properly designed and laid out, and grading includes correctness (with points for Reset, About, and Compute) and form design.

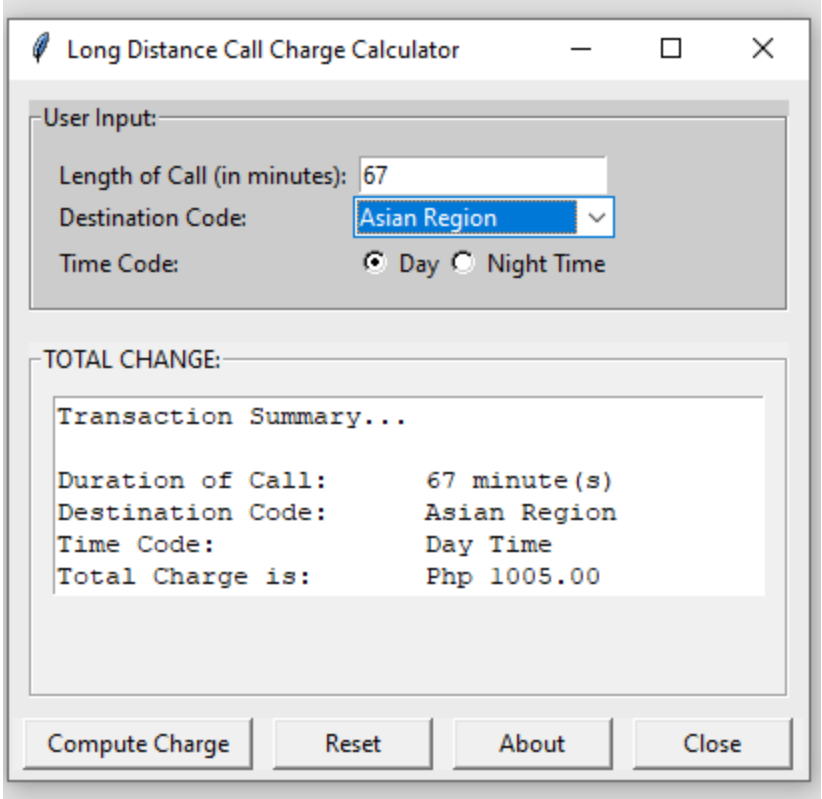
**Code:**

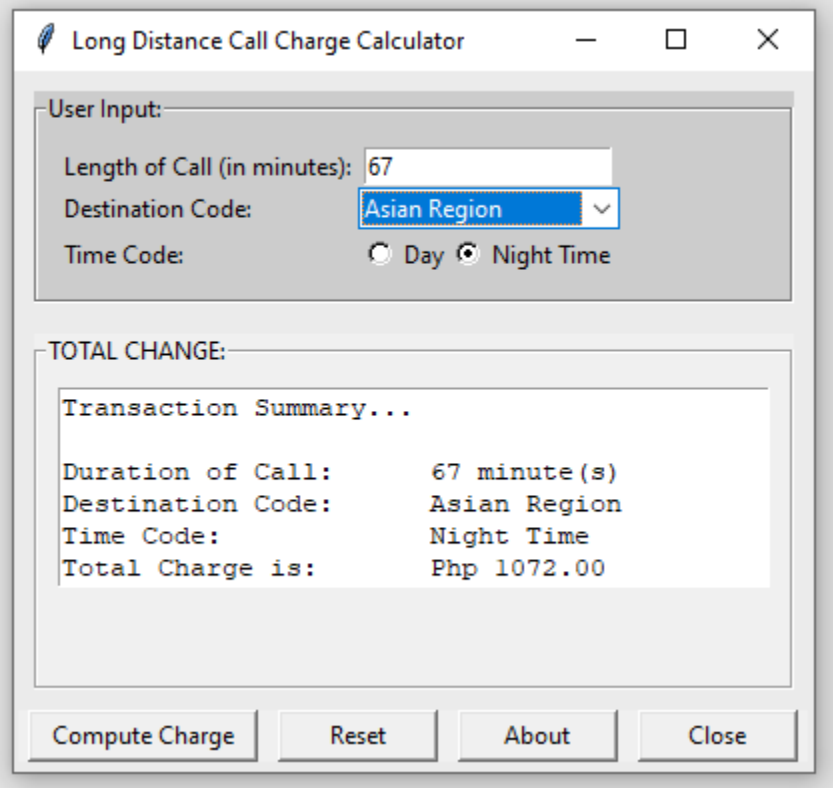
```
import tkinter as tk
from tkinter import ttk, messagebox

class CallChargeApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Long Distance Call Charge Calculator")
        self.root.geometry("400x350")
        self.create_widgets()
        self.load_rates()
        self.root.config(bg='gray92')

    def load_rates(self):
        self.day_rates = {
            "American Region": 50 / 3,
            "Asian Region": 30 / 2,
            "African Region": 40 / 3,
            "European Region": 35 / 2
        }

        self.night_rates = {
            "American Region": 45 / 3,
            "Asian Region": 32 / 2,
            "African Region": 36 / 3,
            "European Region": 30 / 2
        }

    def create_widgets(self):

        frame = tk.LabelFrame(self.root, text="User Input:", bg='gray80',
padx=10, pady=10)
        frame.pack(fill="x", padx=10, pady=10)

        tk.Label(frame, text="Length of Call (in minutes):",
bg='gray80').grid(row=0, column=0, sticky="w")
        self.duration_entry = tk.Entry(frame, width=20)
```

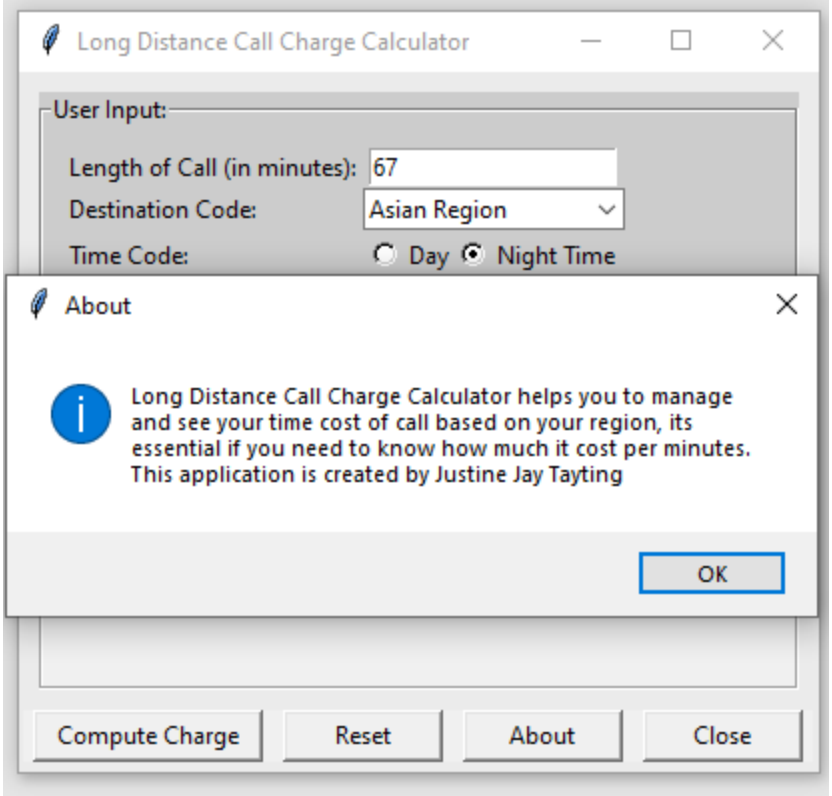
```
        self.duration_entry.grid(row=0, column=1)

        tk.Label(frame, text="Destination Code:", bg='gray80').grid(row=1,
column=0, sticky="w")
        self.destination_cb = ttk.Combobox(frame, width=18, state="readonly")
        self.destination_cb['values'] = ("American Region", "Asian Region",
"African Region", "European Region")
        self.destination_cb.grid(row=1, column=1)
        self.destination_cb.current(0)

        tk.Label(frame, text="Time Code:", bg='gray80').grid(row=2, column=0,
sticky="w")

        self.time_var = tk.StringVar()
        tk.Radiobutton(frame, text="Day Time", variable=self.time_var,
value="Day Time", bg='gray80').grid(row=2, column=1, sticky="w")
        tk.Radiobutton(frame, text="Night Time", variable=self.time_var,
value="Night Time", bg='gray80').grid(row=2, column=1, sticky="e")

        output_frame = tk.LabelFrame(self.root, text="TOTAL CHANGE:",
padx=10, pady=10)
        output_frame.pack(fill="both", expand=True, padx=10, pady=5)

        self.output_text = tk.Text(output_frame, height=6, width=60)
        self.output_text.pack()

        button_frame = tk.Frame(self.root)
        button_frame.pack(pady=5)

        tk.Button(button_frame, text="Compute Charge", width=15,
command=self.compute_charge).grid(row=0, column=0, padx=5)
        tk.Button(button_frame, text="Reset", width=10,
command=self.reset).grid(row=0, column=1, padx=5)
        tk.Button(button_frame, text="About", width=10,
command=self.about).grid(row=0, column=2, padx=5)
        tk.Button(button_frame, text="Close", width=10,
command=self.root.quit).grid(row=0, column=3, padx=5)

    def compute_charge(self):
        try:
            duration = int(self.duration_entry.get())
            if duration <= 0:
                raise ValueError
        except ValueError:
            messagebox.showerror("Error", "Please enter a valid numeric
duration.")
            return

        destination = self.destination_cb.get()
        time_code = self.time_var.get()

        if time_code == "":
            messagebox.showerror("Error", "Please select a time code.")
            return

        if time_code == "Day Time":
            rate = self.day_rates[destination]
```

```
        else:
            rate = self.night_rates[destination]

        total_charge = duration * rate

        self.output_text.delete("1.0", tk.END)
        self.output_text.insert(tk.END, f"Transaction Summary... \n\n")
        self.output_text.insert(tk.END, f"Duration of Call:      {duration}
minute(s)\n")
        self.output_text.insert(tk.END, f"Destination Code:
{destination}\n")
        self.output_text.insert(tk.END, f"Time Code:
{time_code}\n")
        self.output_text.insert(tk.END, f"Total Charge is:       Php
{total_charge:.2f}")

    def reset(self):
        self.duration_entry.delete(0, tk.END)
        self.destination_cb.current(0)
        self.time_var.set("")
        self.output_text.delete("1.0", tk.END)

    def about(self):
        messagebox.showinfo("About", "Long Distance Call Charge Calculator
helps you to manage and see your time cost of call based on your region, its
essential if you need to know how much it cost per minutes. This application
is created by Justine Jay Tayting")

if __name__ == "__main__":
    root = tk.Tk()
    app = CallChargeApp(root)
    root.mainloop()
```

**Output:**


Long Distance Call Charge Calculator
User Input:
Length of Call (in minutes): 67
Destination Code: Asian Region
Time Code: Day Night Time
TOTAL CHANGE:
Transaction Summary...
Duration of Call: 67 minute(s)
Destination Code: Asian Region
Time Code: Day Time
Total Charge is: Php 1005.00
Compute Charge
Reset
About
Close


Long Distance Call Charge Calculator
User Input:
Length of Call (in minutes): 67
Destination Code: Asian Region
Time Code: Day Night Time
TOTAL CHANGE:
Transaction Summary...
Duration of Call: 67 minute(s)
Destination Code: Asian Region
Time Code: Night Time
Total Charge is: Php 1072.00
Compute Charge
Reset
About
Close


Long Distance Call Charge Calculator
User Input:
Length of Call (in minutes): 67
Destination Code: Asian Region
Time Code: Day Night Time
About
Long Distance Call Charge Calculator helps you to manage and see your time cost of call based on your region, its essential if you need to know how much it cost per minutes. This application is created by Justine Jay Tayting
OK
Compute Charge
Reset
About
Close