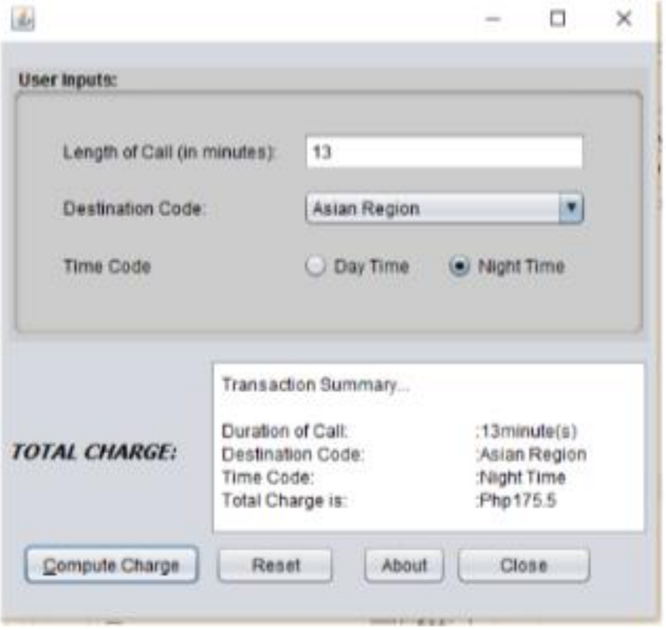
Reyes, Maurey Shane M.

BSCS-C204

# Finals Lab Task 4. Python GUI using TKINTER

7. See sample output below:



User Inputs:

Length of Call (in minutes): 13

Destination Code: Asian Region

Time Code ○ Day Time ● Night Time

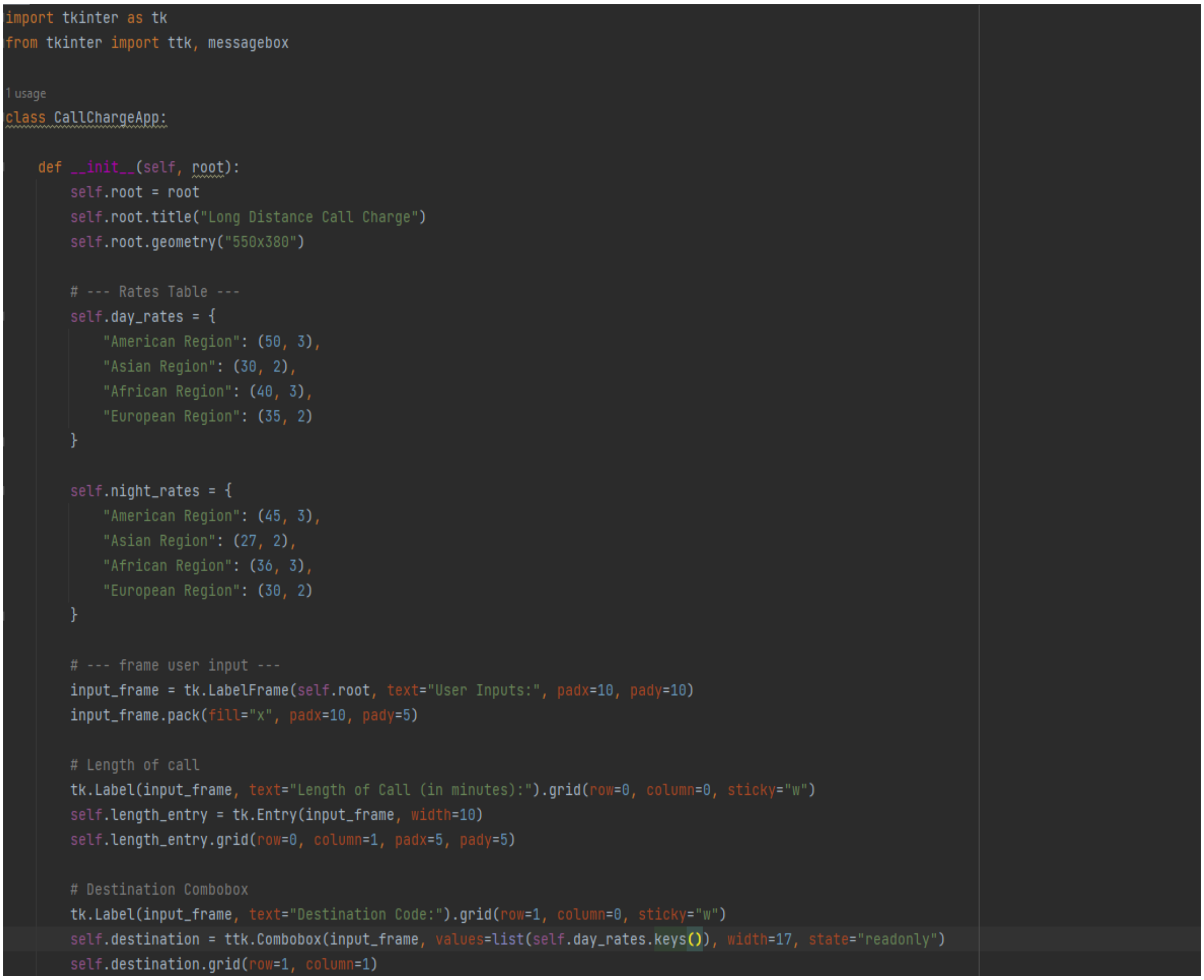
TOTAL CHARGE:

```
Transaction Summary...

Duration of Call:          :13minute(s)
Destination Code:          :Asian Region
Time Code:                 :Night Time
Total Charge is:           :Php175.5
```

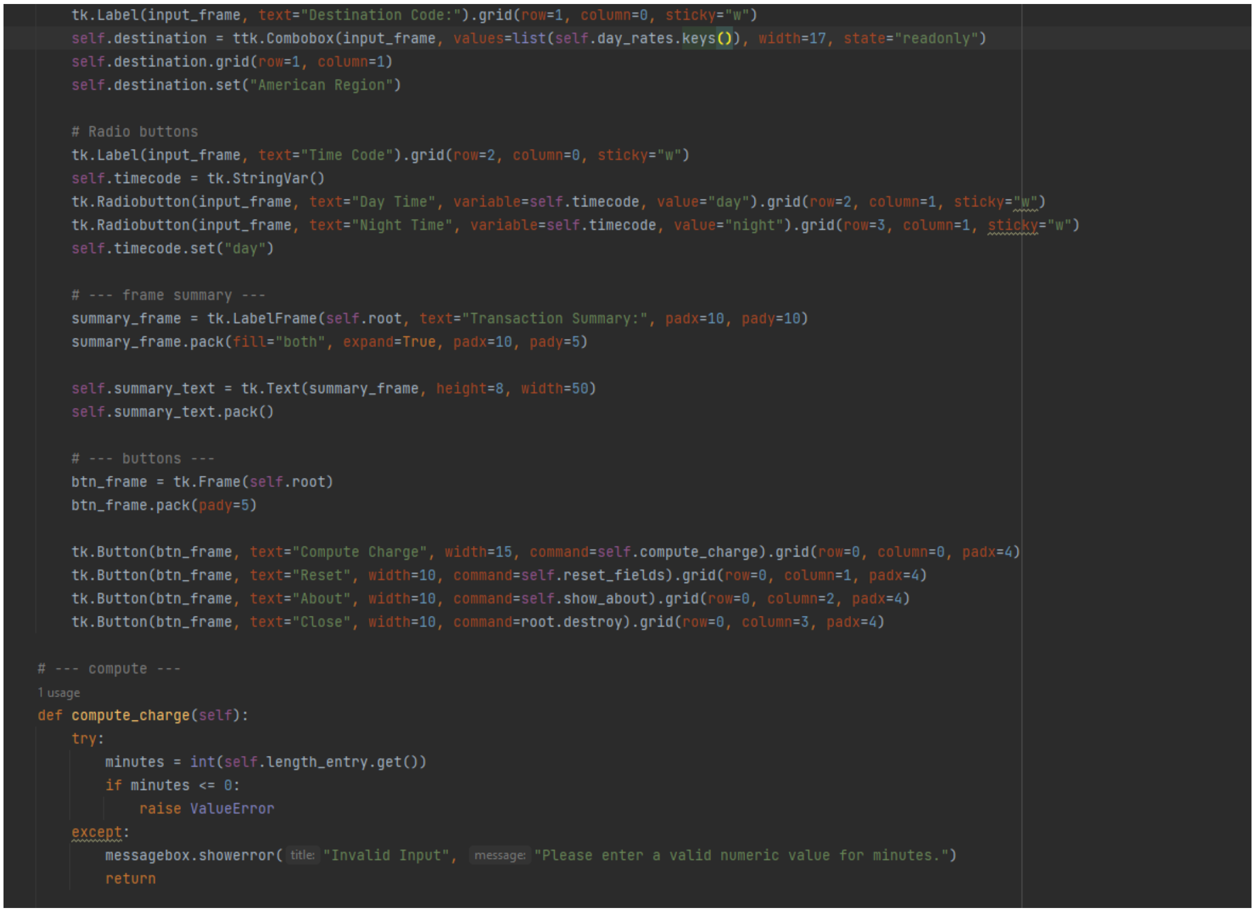
**Rubrics: Form Design and Layout : 10 points**

**Program Correctness : 40 points (Reset – 5 pts., About – 5 pts. , Compute – 30 pts.)**

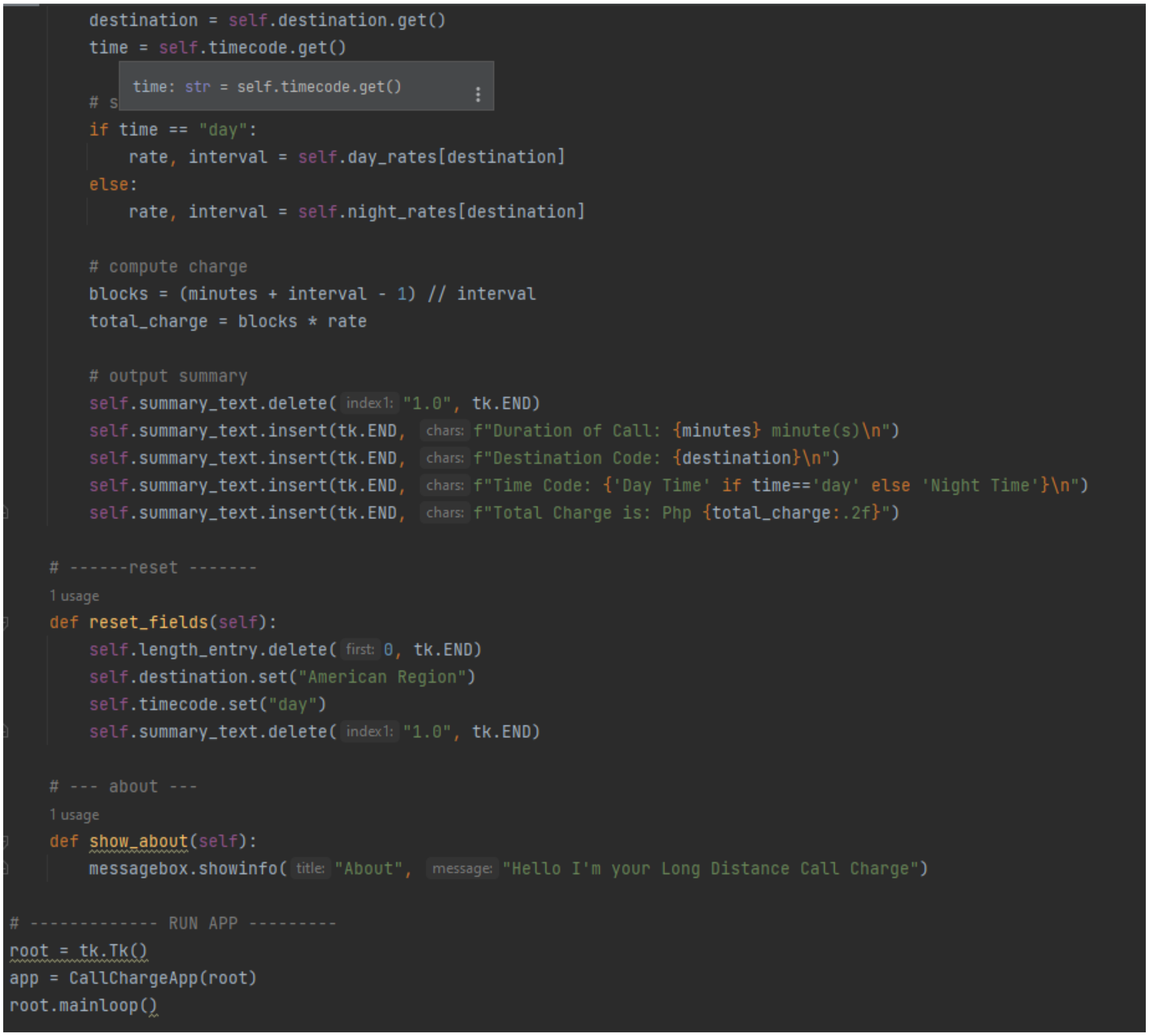
# Source Code:



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

1 usage
class CallChargeApp:

    def __init__(self, root):
        self.root = root
        self.root.title("Long Distance Call Charge")
        self.root.geometry("550x380")

        # --- Rates Table ---
        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": (27, 2),
            "African Region": (36, 3),
            "European Region": (30, 2)
        }

        # --- frame user input ---
        input_frame = tk.LabelFrame(self.root, text="User Inputs:", padx=10, pady=10)
        input_frame.pack(fill="x", padx=10, pady=5)

        # Length of call
        tk.Label(input_frame, text="Length of Call (in minutes):").grid(row=0, column=0, sticky="w")
        self.length_entry = tk.Entry(input_frame, width=10)
        self.length_entry.grid(row=0, column=1, padx=5, pady=5)

        # Destination Combobox
        tk.Label(input_frame, text="Destination Code:").grid(row=1, column=0, sticky="w")
        self.destination = ttk.Combobox(input_frame, values=list(self.day_rates.keys()), width=17, state="readonly")
        self.destination.grid(row=1, column=1)
```



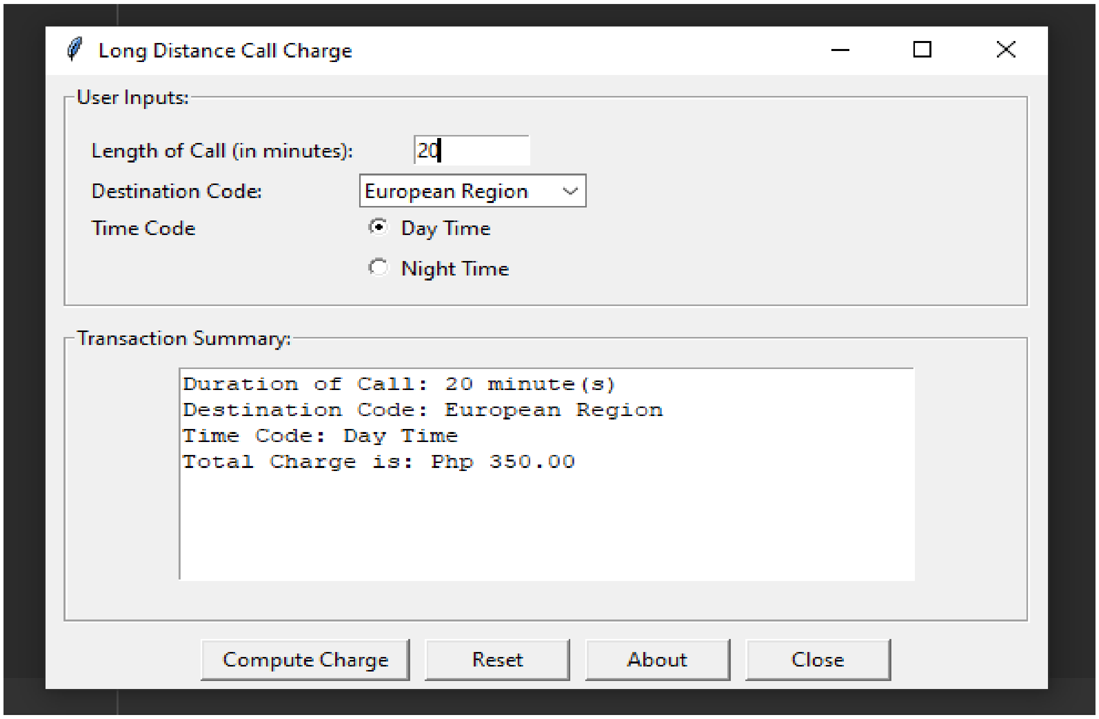
```
        tk.Label(input_frame, text="Destination Code:").grid(row=1, column=0, sticky="w")
        self.destination = ttk.Combobox(input_frame, values=list(self.day_rates.keys()), width=17, state="readonly")
        self.destination.grid(row=1, column=1)
        self.destination.set("American Region")

        # Radio buttons
        tk.Label(input_frame, text="Time Code").grid(row=2, column=0, sticky="w")
        self.timecode = tk.StringVar()
        tk.Radiobutton(input_frame, text="Day Time", variable=self.timecode, value="day").grid(row=2, column=1, sticky="w")
        tk.Radiobutton(input_frame, text="Night Time", variable=self.timecode, value="night").grid(row=3, column=1, sticky="w")
        self.timecode.set("day")

        # --- frame summary ---
        summary_frame = tk.LabelFrame(self.root, text="Transaction Summary:", padx=10, pady=10)
        summary_frame.pack(fill="both", expand=True, padx=10, pady=5)

        self.summary_text = tk.Text(summary_frame, height=8, width=50)
        self.summary_text.pack()

        # --- buttons ---
        btn_frame = tk.Frame(self.root)
        btn_frame.pack(pady=5)

        tk.Button(btn_frame, text="Compute Charge", width=15, command=self.compute_charge).grid(row=0, column=0, padx=4)
        tk.Button(btn_frame, text="Reset", width=10, command=self.reset_fields).grid(row=0, column=1, padx=4)
        tk.Button(btn_frame, text="About", width=10, command=self.show_about).grid(row=0, column=2, padx=4)
        tk.Button(btn_frame, text="Close", width=10, command=root.destroy).grid(row=0, column=3, padx=4)

    # --- compute ---
    1 usage
    def compute_charge(self):
        try:
            minutes = int(self.length_entry.get())
            if minutes <= 0:
                raise ValueError
        except:
            messagebox.showerror( title: "Invalid Input",  message: "Please enter a valid numeric value for minutes.")
            return
```



```
        destination = self.destination.get()
        time = self.timecode.get()
              time: str = self.timecode.get()
        # s
        if time == "day":
            rate, interval = self.day_rates[destination]
        else:
            rate, interval = self.night_rates[destination]

        # compute charge
        blocks = (minutes + interval - 1) // interval
        total_charge = blocks * rate

        # output summary
        self.summary_text.delete( index1: "1.0", tk.END)
        self.summary_text.insert(tk.END,  chars: f"Duration of Call: {minutes} minute(s)\n")
        self.summary_text.insert(tk.END,  chars: f"Destination Code: {destination}\n")
        self.summary_text.insert(tk.END,  chars: f"Time Code: {'Day Time' if time=='day' else 'Night Time'}\n")
        self.summary_text.insert(tk.END,  chars: f"Total Charge is: Php {total_charge:.2f}")

    # ------reset -------
    1 usage
    def reset_fields(self):
        self.length_entry.delete( first: 0, tk.END)
        self.destination.set("American Region")
        self.timecode.set("day")
        self.summary_text.delete( index1: "1.0", tk.END)

    # --- about ---
    1 usage
    def show_about(self):
        messagebox.showinfo( title: "About",  message: "Hello I'm your Long Distance Call Charge")

# ------------- RUN APP ---------
root = tk.Tk()
app = CallChargeApp(root)
root.mainloop()
```

## Sample Output:

## Day Time

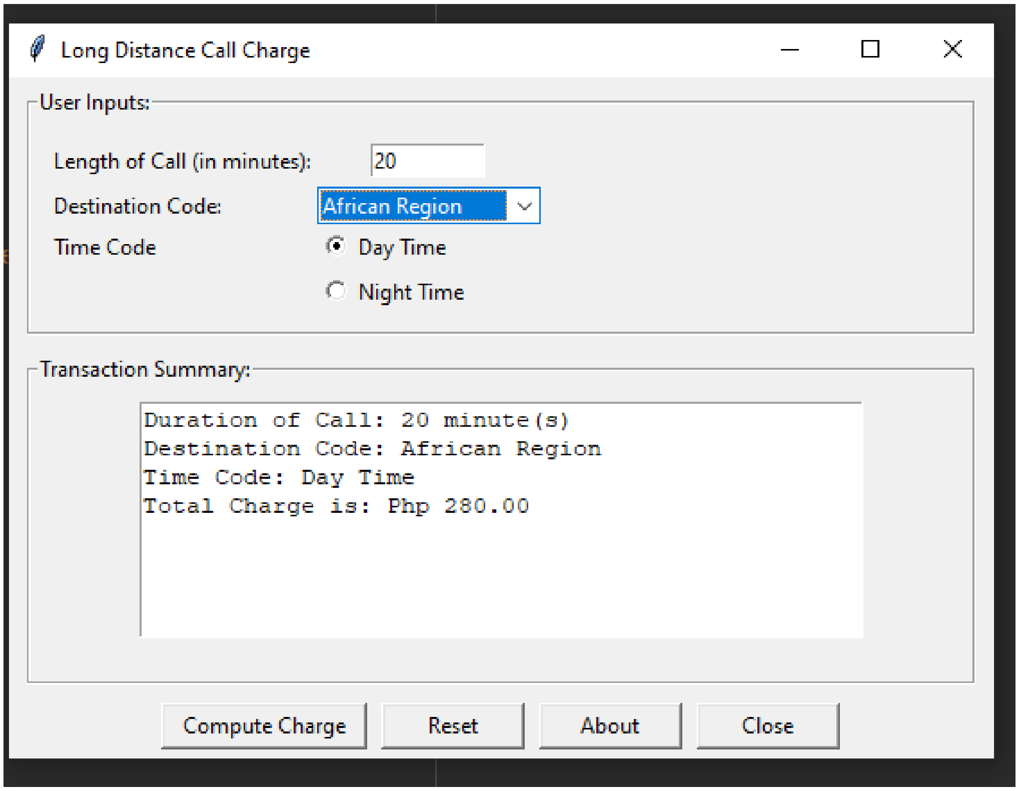


Length of Call (in minutes): 20

Destination Code: European Region

Time Code ● Day Time ○ Night Time

```
Duration of Call: 20 minute(s)
Destination Code: European Region
Time Code: Day Time
Total Charge is: Php 350.00
```

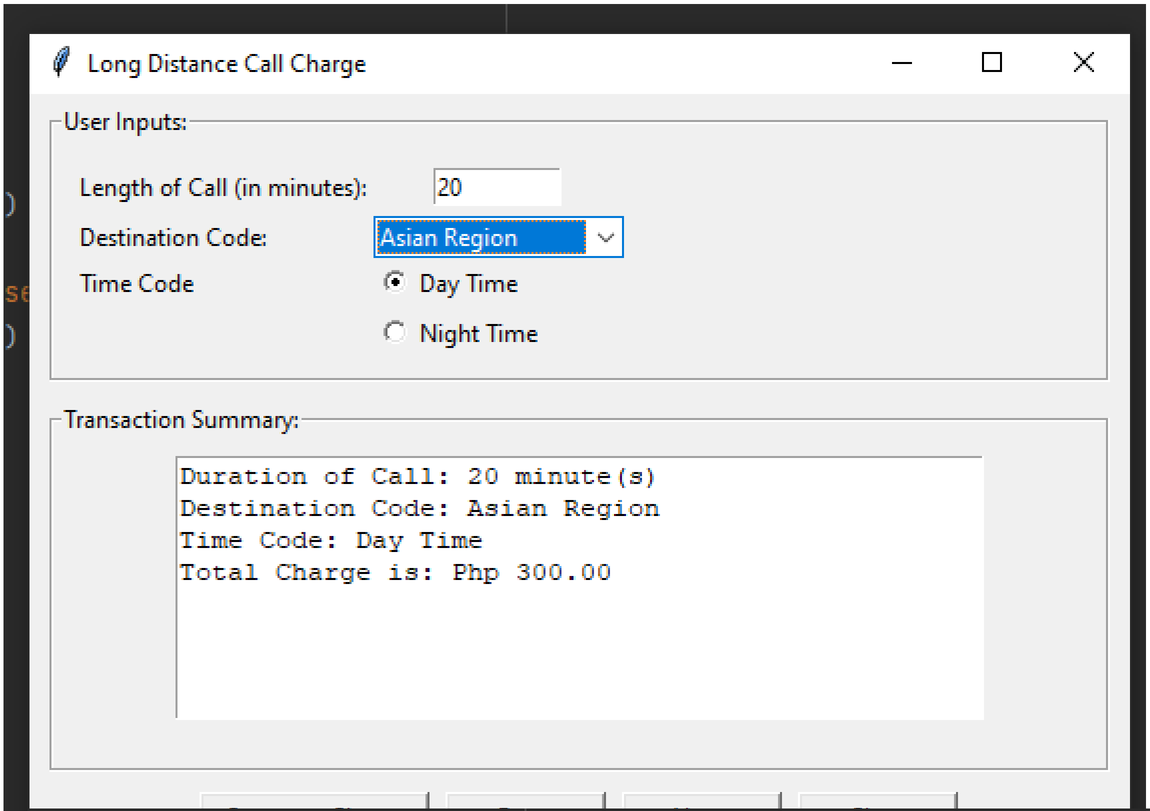


Length of Call (in minutes): 20

Destination Code: African Region

Time Code ● Day Time ○ Night Time

```
Duration of Call: 20 minute(s)
Destination Code: African Region
Time Code: Day Time
Total Charge is: Php 280.00
```

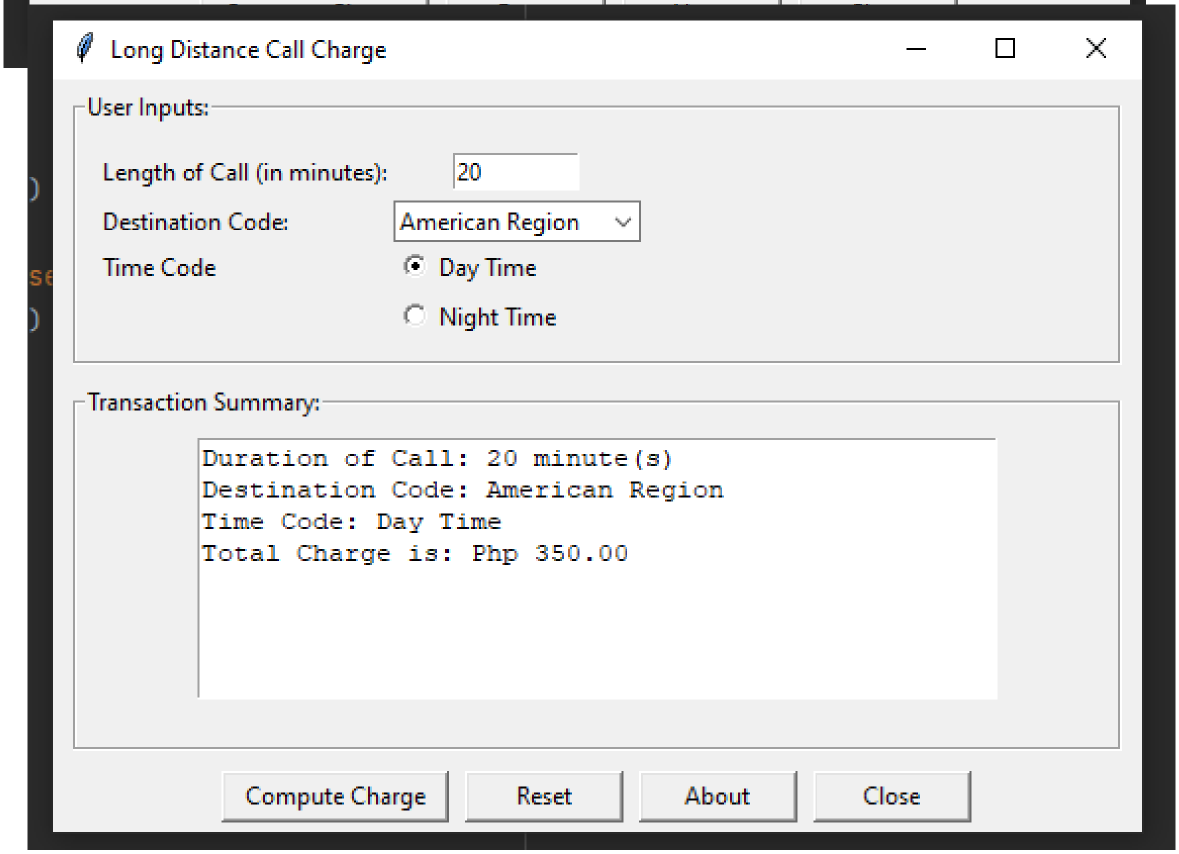


Length of Call (in minutes): 20

Destination Code: Asian Region

Time Code ● Day Time ○ Night Time

```
Duration of Call: 20 minute(s)
Destination Code: Asian Region
Time Code: Day Time
Total Charge is: Php 300.00
```



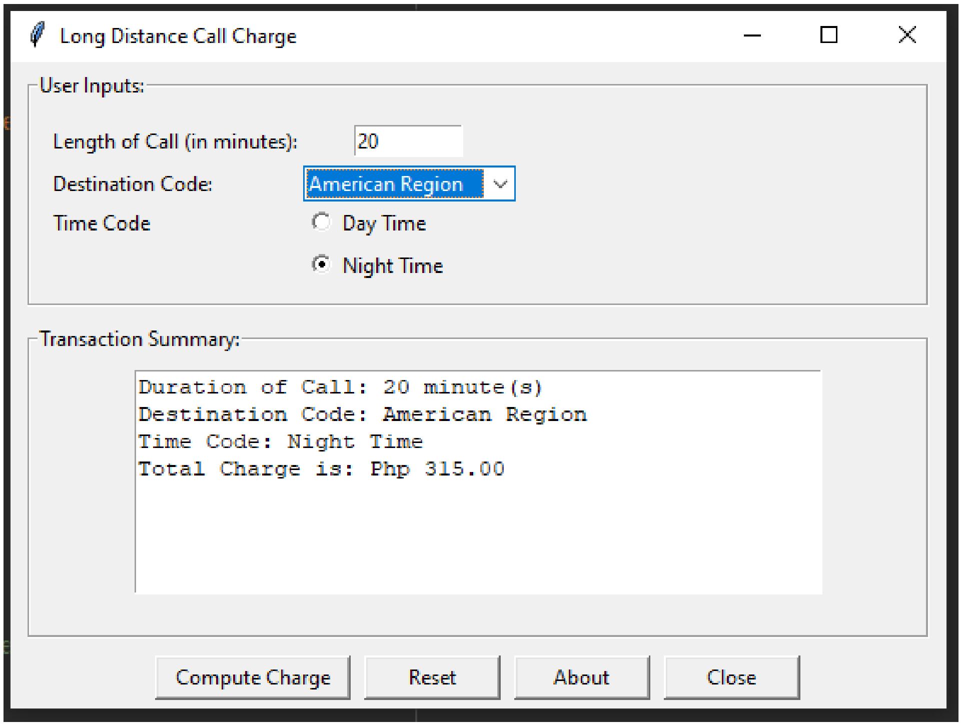
Length of Call (in minutes): 20

Destination Code: American Region

Time Code ● Day Time ○ Night Time

```
Duration of Call: 20 minute(s)
Destination Code: American Region
Time Code: Day Time
Total Charge is: Php 350.00
```

## Night time:

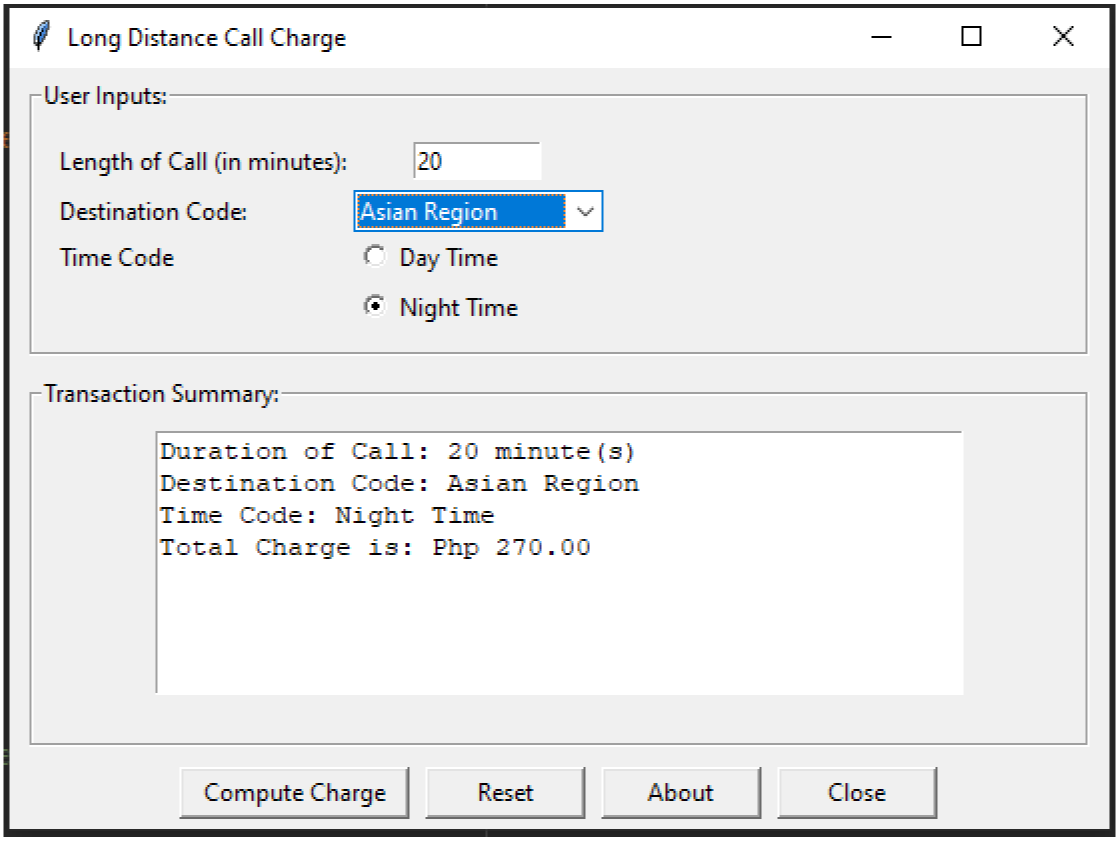


Length of Call (in minutes): 20

Destination Code: American Region

Time Code ○ Day Time ● Night Time

```
Duration of Call: 20 minute(s)
Destination Code: American Region
Time Code: Night Time
Total Charge is: Php 315.00
```

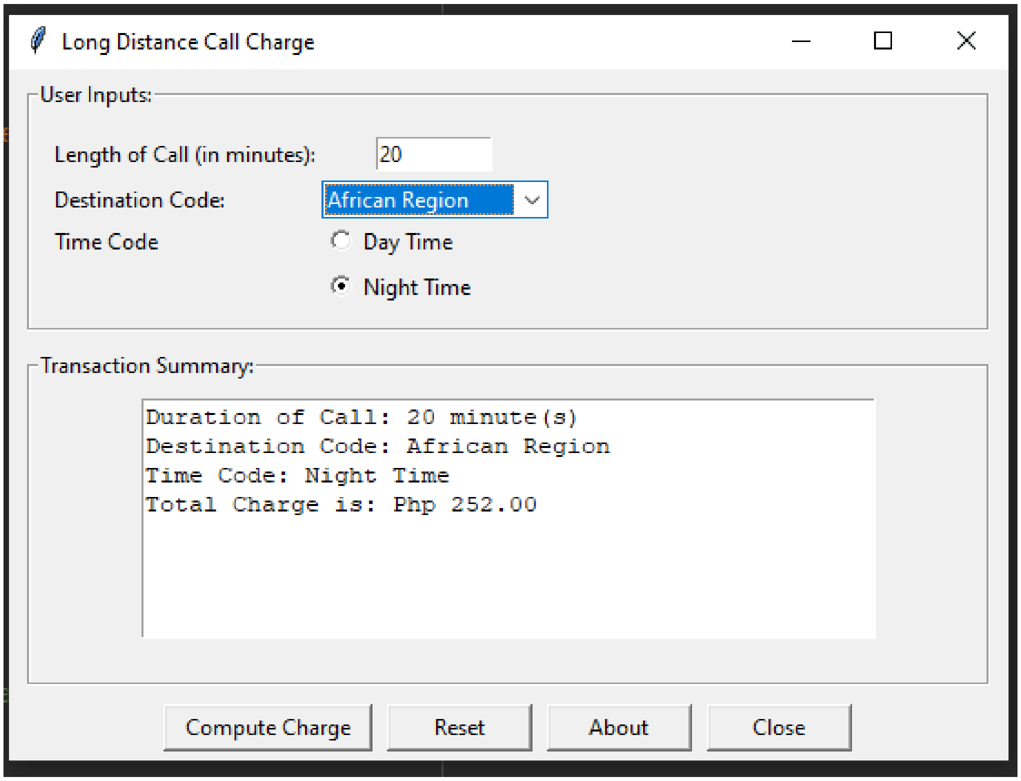


Length of Call (in minutes): 20

Destination Code: Asian Region

Time Code ○ Day Time ● Night Time

```
Duration of Call: 20 minute(s)
Destination Code: Asian Region
Time Code: Night Time
Total Charge is: Php 270.00
```



User Inputs:

Length of Call (in minutes): 20

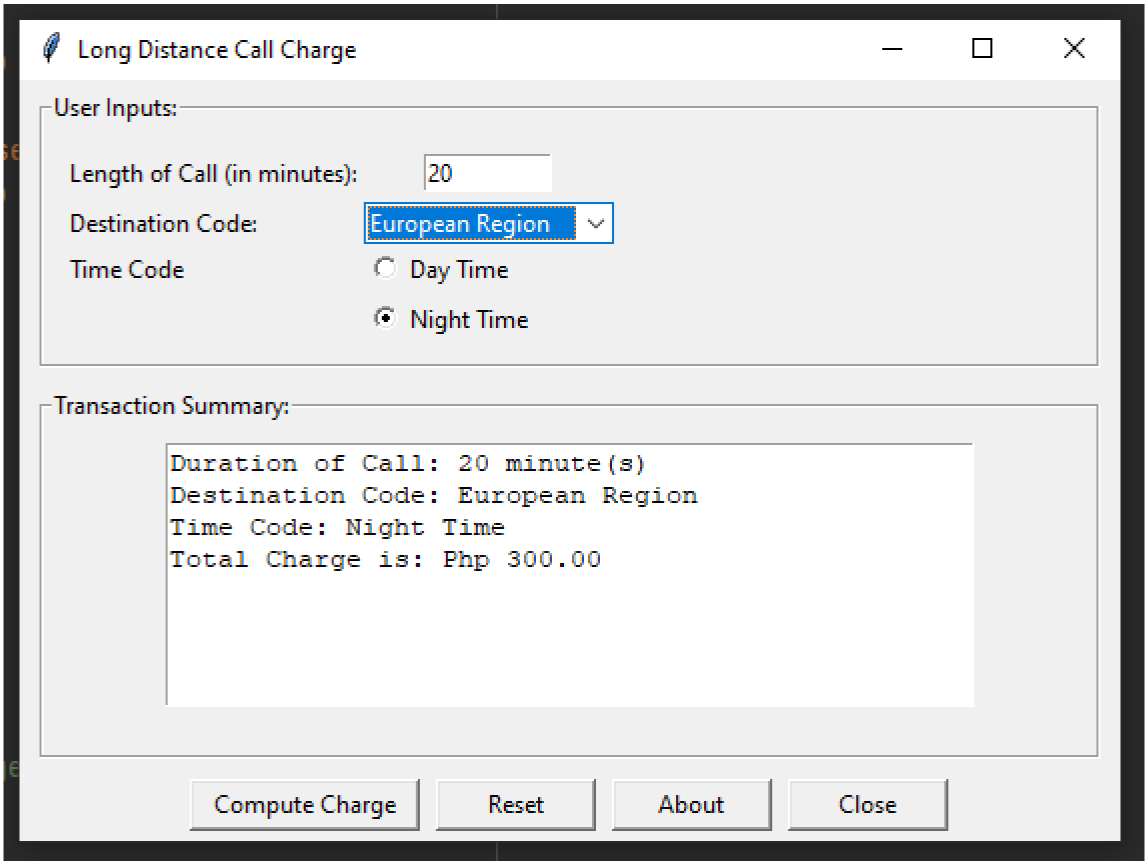
Destination Code: African Region

Time Code ○ Day Time

● Night Time

Transaction Summary:

```
Duration of Call: 20 minute(s)
Destination Code: African Region
Time Code: Night Time
Total Charge is: Php 252.00
```



User Inputs:

Length of Call (in minutes): 20

Destination Code: European Region

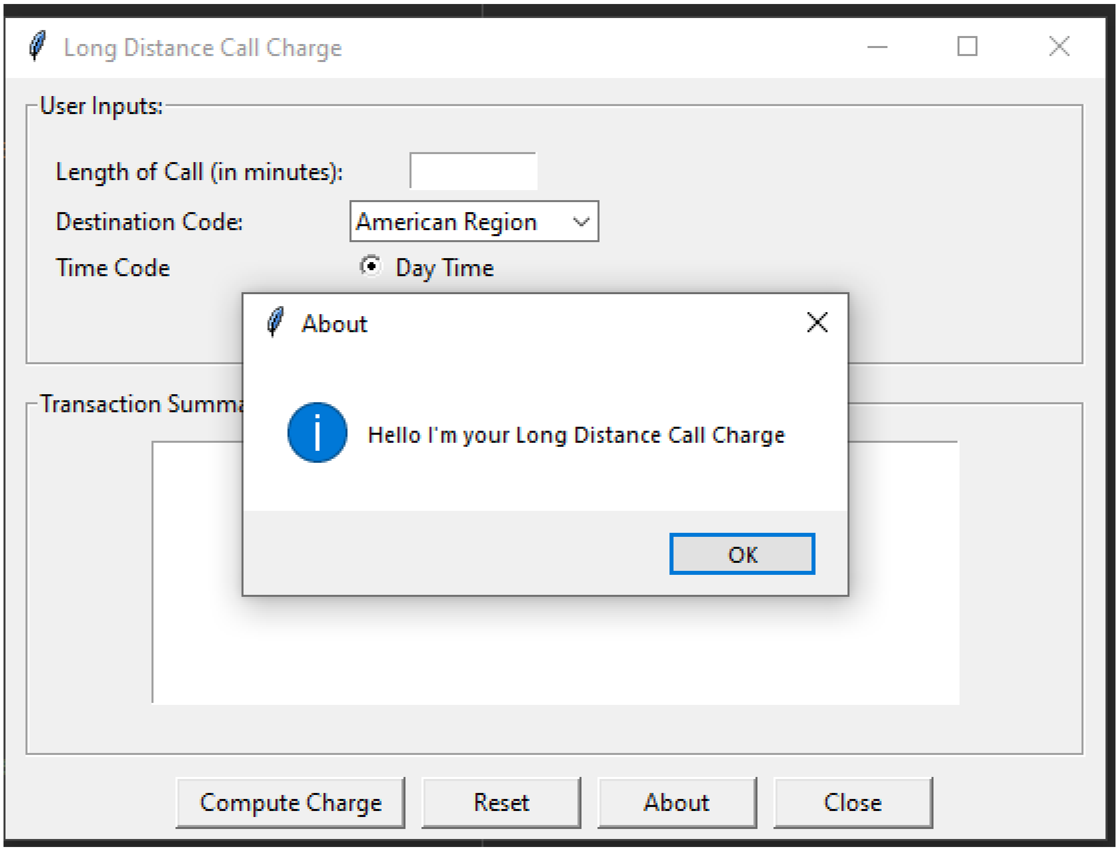
Time Code ○ Day Time

● Night Time

Transaction Summary:

```
Duration of Call: 20 minute(s)
Destination Code: European Region
Time Code: Night Time
Total Charge is: Php 300.00
```

## About:



Length of Call (in minutes):

Destination Code: American Region

Time Code ● Day Time

Hello I'm your Long Distance Call Charge