Torres, Justine Kurt Q.
7OOP
BSCS – C204
Finals Task 4

Python and Tkinter GUI Program

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**.

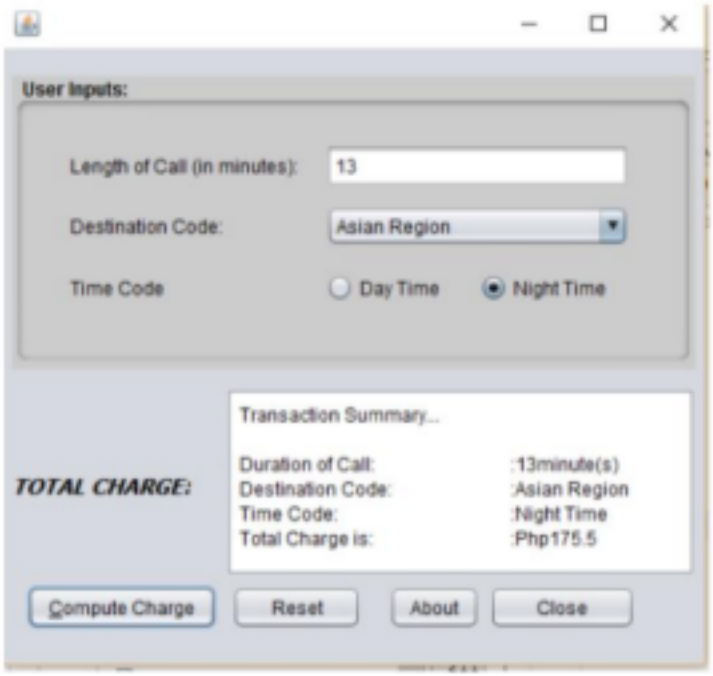
   4.1 Computations should be based on the table rates shown above. (The total charge is based on **Length of Calls**, **Destination Code** and **Time Code**)

   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:



Rubrics: Form Design and Layout : 10 points

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

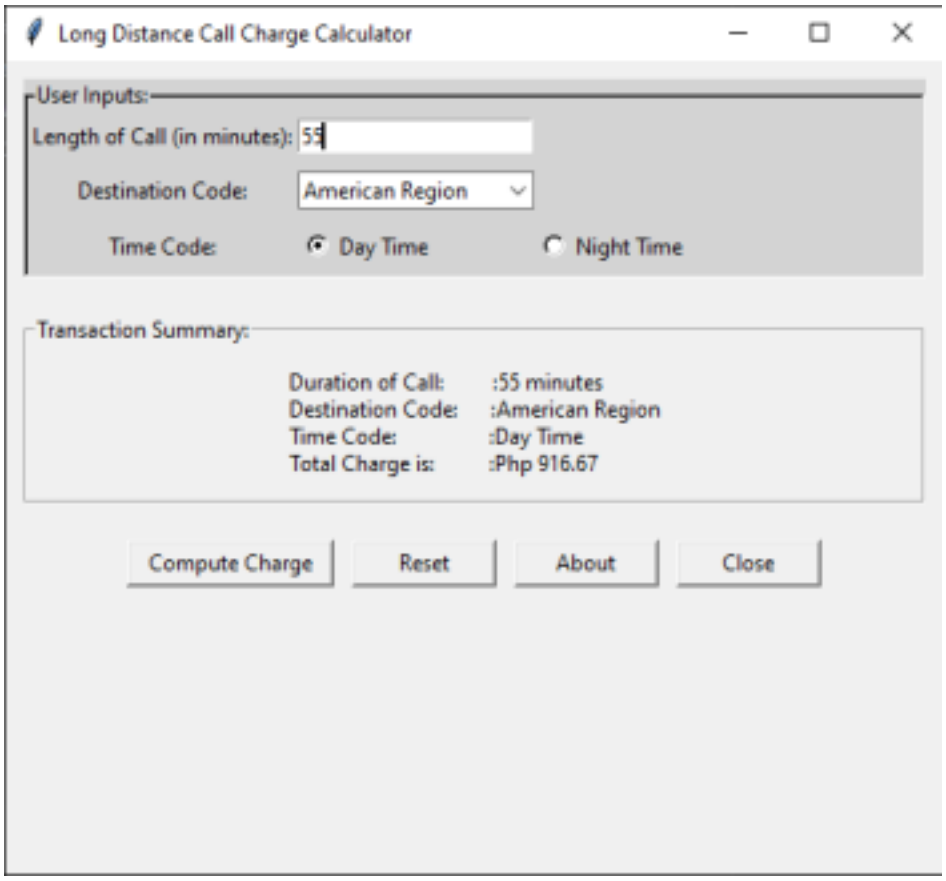
Sample Code:

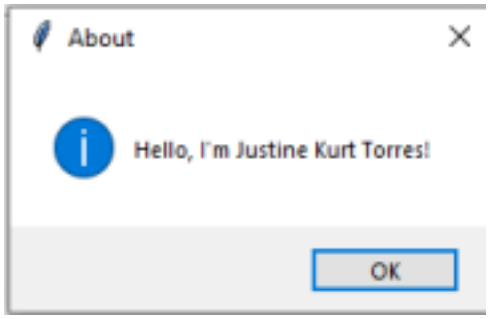
```
import tkinter as tk
from tkinter import ttk, messagebox


class LongDistanceCallGUI:
 def __init__(self, master):
 self.master = master
 self.master.title("Long Distance Call Charge Calculator")
self.master.geometry("520x450")

 self.day_rates = {
 "American Region": 50,
 "Asian Region": 30,
 "African Region": 40,
 "European Region": 35
 }

 self.night_rates = {
 "American Region": 45,
 "Asian Region": 27,
 "African Region": 36,
 "European Region": 30
 }
```

```
 self.create_widgets()
 def create_widgets(self):
 frame = tk.LabelFrame(self.master, text="User Inputs:",
relief='sunken', bg='lightgray')
 frame.pack(padx=10, pady=10, fill="both")

 tk.Label(frame, text="Length of Call (in minutes):",
bg='lightgray').grid(row=0, column=2, pady=2)
 self.entry_minutes = tk.Entry(frame, width=21)
 self.entry_minutes.grid(row=0, column=3, pady=5)

 tk.Label(frame, text="Destination Code:", bg='lightgray').grid(row=1,
column=2, pady=5)
 self.dest_var = tk.StringVar()
 self.combo_dest = ttk.Combobox(frame, width=18,
textvariable=self.dest_var, state="readonly")
 self.combo_dest['values'] = ("American Region", "Asian Region",
"African Region", "European Region")
 self.combo_dest.grid(row=1, column=3, pady=5)
 self.combo_dest.current(0)

 tk.Label(frame, text="Time Code:", bg='lightgray').grid(row=2,
column=2, pady=5)
 self.time_var = tk.StringVar()
 tk.Radiobutton(frame, text="Day Time", variable=self.time_var,
value="day", bg='lightgray').grid(row=2, column=3, sticky='w')
tk.Radiobutton(frame, text="Night Time", variable=self.time_var,
value="night", bg='lightgray').grid(row=2, column=4, sticky='w')
self.time_var.set("day")


 self.output_frame = tk.LabelFrame(self.master, text="Transaction
Summary:")
 self.output_frame.pack(padx=10, pady=10, fill="both")

 self.lbl_summary = tk.Label(self.output_frame, justify="left",
anchor="w")
 self.lbl_summary.pack(padx=10, pady=10)


 btn_frame = tk.Frame(self.master)
 btn_frame.pack(pady=10)

 tk.Button(btn_frame, text="Compute Charge", width=15,
command=self.compute_charge).grid(row=0, column=0, padx=5)
 tk.Button(btn_frame, text="Reset", width=10,
command=self.reset_all).grid(row=0, column=1, padx=5)
 tk.Button(btn_frame, text="About", width=10,
command=self.show_about).grid(row=0, column=2, padx=5)
 tk.Button(btn_frame, text="Close", width=10,
command=self.master.quit).grid(row=0, column=3, padx=5)


 def compute_charge(self):
 try:
```

```
 minutes = int(self.entry_minutes.get())
 if minutes <= 0:
 raise ValueError
 except ValueError:
 messagebox.showerror("Invalid Input", "Please enter a valid  positive
number for minutes.")
 return

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

 if time_code == "day":
 rate = self.day_rates[destination]
 time_display = "Day Time"
 else:
 rate = self.night_rates[destination]
 time_display = "Night Time"
 charge = (minutes / 3) * rate

 summary_text = (
 f"Duration of Call: :{minutes} minutes\n"  f"Destination
Code: :{destination}\n"  f"Time Code: :{time_display}\n"
f"Total Charge is: :Php {charge:.2f}"  )

 self.lbl_summary.config(text=summary_text)


 def reset_all(self):
 self.entry_minutes.delete(0, tk.END)
 self.combo_dest.current(0)
 self.time_var.set("day")
 self.lbl_summary.config(text="")


 def show_about(self):
 messagebox.showinfo("About", "Hello, I'm Justine Kurt Torres!")


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

Output:


Long Distance Call Charge Calculator
User Inputs:
Length of Call (in minutes): 55
Destination Code: American Region
Time Code: Day Time Night Time
Transaction Summary:
Duration of Call: :55 minutes
Destination Code: :American Region
Time Code: :Day Time
Total Charge is: :Php 916.67
Compute Charge
Reset
About
Close


About
Hello, I'm Justine Kurt Torres!
OK