Ogania, Johnest Wheyne D.
BSCS-C204
7OOP

Finals Task 4. Python and Tkinter GUI program

Problem:

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		NIGHTTIME 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.**

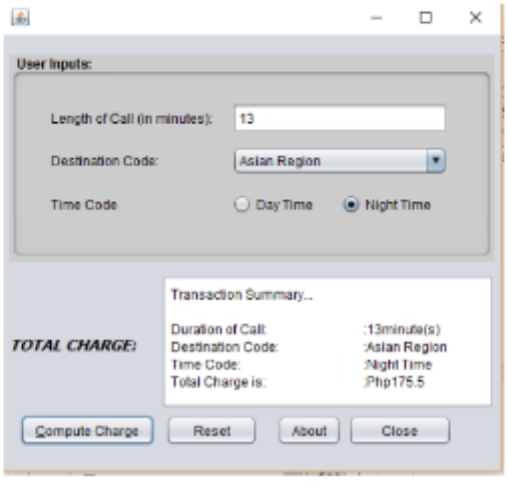
   4.1 Computations should be based on the table rates shown above. (The total charge is based on **Length of Calls**, **Destination Code** and **Time Code**)

   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.)

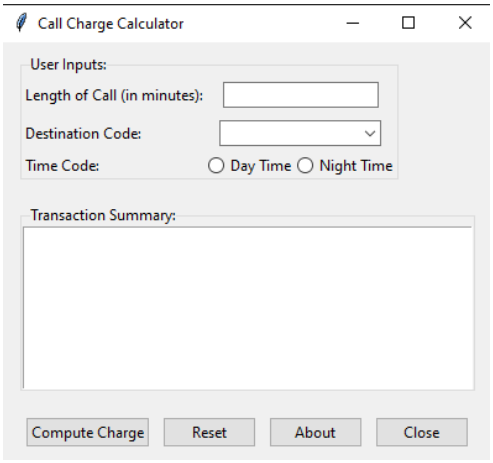
Sample Code:

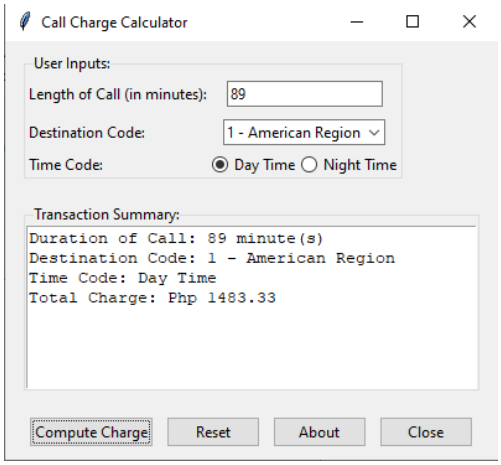
```python
import tkinter as tk
from tkinter import ttk, messagebox

1 usage
class CallChargeCalculator:
    RATES = {
        1: (50, 45),
        2: (20, 27),
        3: (40, 36),
        4: (35, 30)
    }

    def __init__(self, root):
        self.root = root
        self.root.title("Call Charge Calculator")

        self.time_code = tk.StringVar()

        self.create_widgets()

    1 usage
    def create_widgets(self):
        frame_input = ttk.LabelFrame(self.root, text="User Inputs:")
        frame_input.grid(row=0, column=0, padx=15, pady=10, sticky="w")

        ttk.Label(frame_input, text="Length of Call (in minutes):").grid(row=0, column=0, sticky="w")
        self.entry_minutes = ttk.Entry(frame_input, width=20)
        self.entry_minutes.grid(row=0, column=1, pady=5)

        ttk.Label(frame_input, text="Destination Code:").grid(row=1, column=0, sticky="w")

        self.combo_dest = ttk.Combobox(frame_input, state="readonly", width=18)
        self.combo_dest['values'] = [
            "1 - American Region",
            "2 - Asian Region",
            "3 - African Region",
            "4 - European Region"
        ]
```

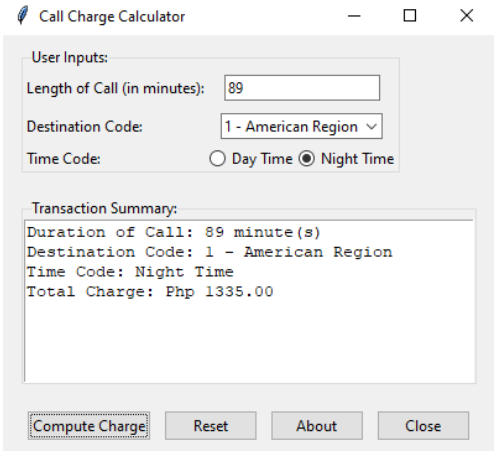
```
        self.combo_dest.grid(row=1, column=1, pady=5)

        ttk.Label(frame_input, text="Time Code:").grid(row=2, column=0, sticky="w")

        rb_frame = ttk.Frame(frame_input)
        rb_frame.grid(row=2, column=1, sticky="w")

        ttk.Radiobutton(rb_frame, text="Day Time", variable=self.time_code, value="day").pack(side="left")
        ttk.Radiobutton(rb_frame, text="Night Time", variable=self.time_code, value="night").pack(side="left")

        frame_summary = ttk.LabelFrame(self.root, text="Transaction Summary:")
        frame_summary.grid(row=1, column=0, padx=15, pady=10, sticky="w")

        self.summary_text = tk.Text(frame_summary, width=45, height=8, state='disabled')
        self.summary_text.pack()

        frame_buttons = ttk.Frame(self.root)
        frame_buttons.grid(row=2, column=0, pady=10)

        ttk.Button(frame_buttons, text="Compute Charge", command=self.compute).grid(row=0, column=0, padx=5)
        ttk.Button(frame_buttons, text="Reset", command=self.reset).grid(row=0, column=1, padx=5)
        ttk.Button(frame_buttons, text="About", command=self.show_about).grid(row=0, column=2, padx=5)
        ttk.Button(frame_buttons, text="Close", command=self.root.destroy).grid(row=0, column=3, padx=5)

    1 usage
    def compute(self):
        try:
            minutes = int(self.entry_minutes.get())
            if minutes <= 0:
                raise ValueError

        except ValueError:
            messagebox.showerror( title: "Invalid Input",  message: "Please enter a valid numeric value for minutes.")
            return

        if not self.combo_dest.get():
            messagebox.showwarning( title: "Missing Field",  message: "Please select a Destination Code.")
            return
```

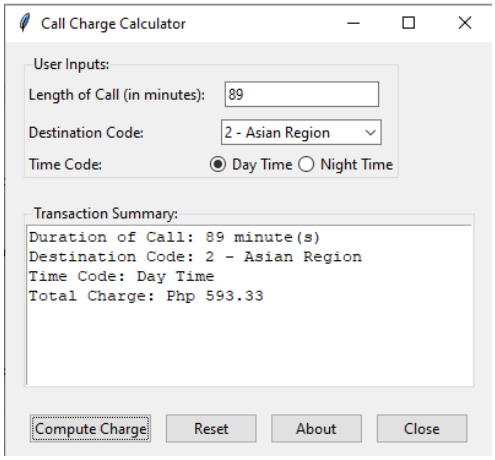
```
        if not self.time_code.get():
            messagebox.showwarning( title: "Missing Field",  message: "Please select a Time Code.")
            return

        dest_code = int(self.combo_dest.get().split(" ")[0])
        rate = self.RATES[dest_code][0] if self.time_code.get() == "day" else self.RATES[dest_code][1]
        blocks_3min = minutes / 3
        total_charge = blocks_3min * rate

        self.summary_text.config(state='normal')
        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: {self.combo_dest.get()}\n")
        self.summary_text.insert(tk.END,  chars: f"Time Code: {'Day Time' if self.time_code.get()=='day' else 'Night Time'}\n")
        self.summary_text.insert(tk.END,  chars: f"Total Charge: Php {total_charge:.2f}\n")
        self.summary_text.config(state='disabled')

    1 usage
    def reset(self):
        self.entry_minutes.delete( first: 0, tk.END)
        self.combo_dest.set('')
        self.time_code.set('')
        self.summary_text.config(state='normal')
        self.summary_text.delete( index1: 1.0, tk.END)
        self.summary_text.config(state='disabled')

    1 usage
    def show_about(self):
        messagebox.showinfo( title: "About",  message: "Hello I'm Your Call Charge Calculator")

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

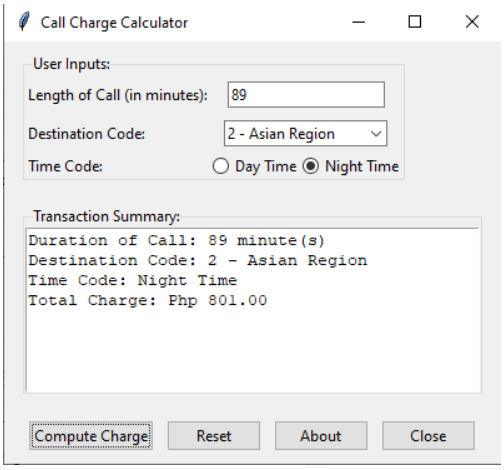
Sample Output:

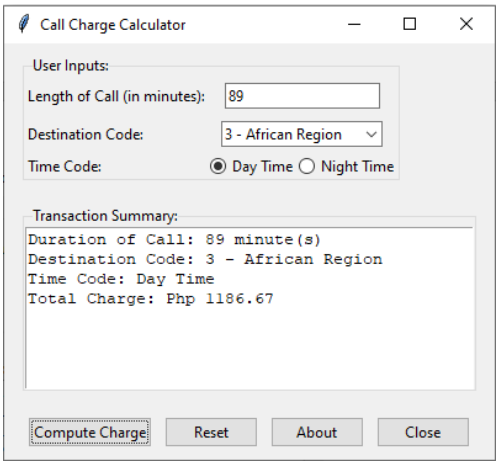


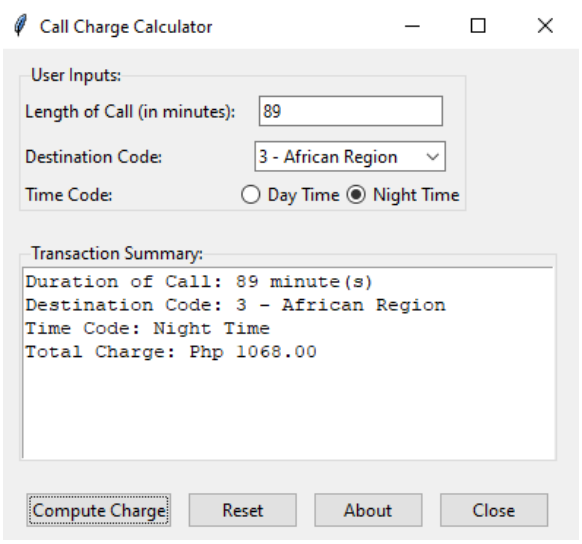


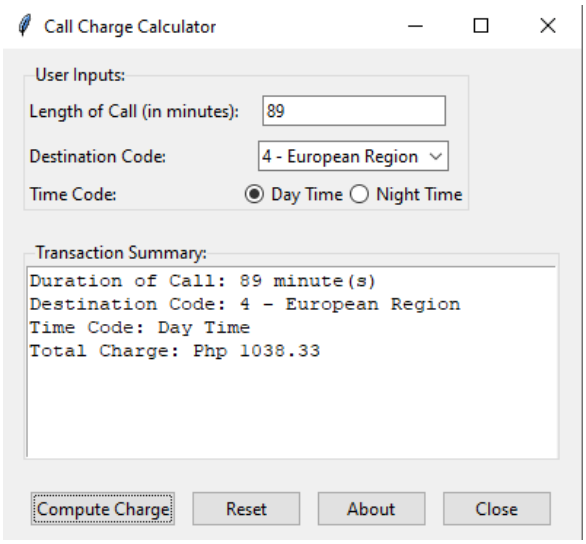


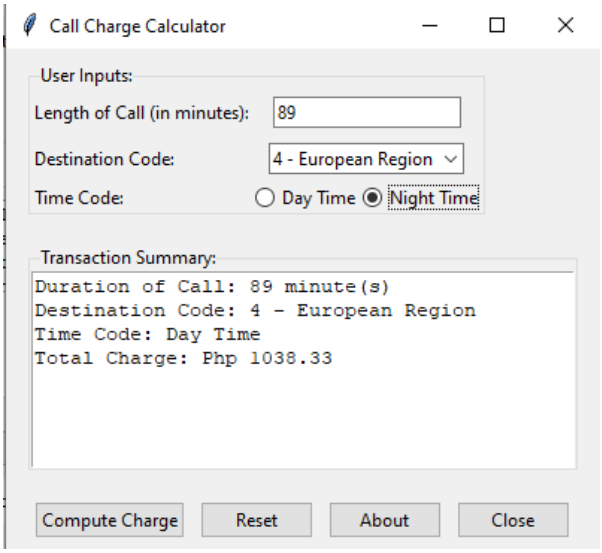


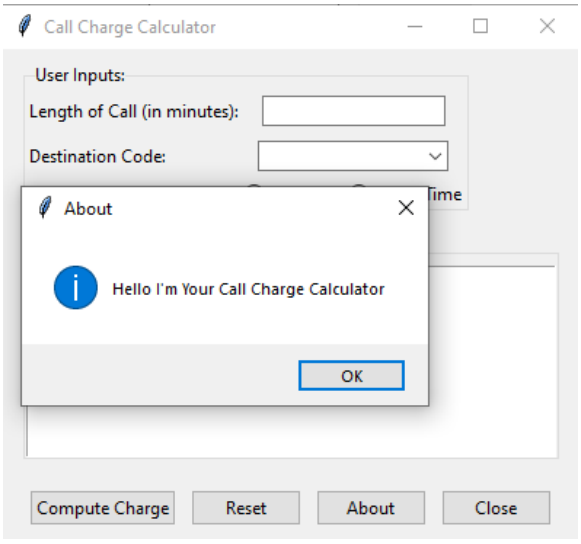





Call Charge Calculator
User Inputs:
Length of Call (in minutes):
89
Destination Code:
3 - African Region
Time Code:
Day Time
Night Time
Transaction Summary:
Duration of Call: 89 minute(s)
Destination Code: 3 - African Region
Time Code: Night Time
Total Charge: Php 1068.00
Compute Charge
Reset
About
Close


Call Charge Calculator
User Inputs:
Length of Call (in minutes):
89
Destination Code:
4 - European Region
Time Code:
Day Time
Night Time
Transaction Summary:
Duration of Call: 89 minute(s)
Destination Code: 4 - European Region
Time Code: Day Time
Total Charge: Php 1038.33
Compute Charge
Reset
About
Close


Call Charge Calculator
User Inputs:
Length of Call (in minutes):
89
Destination Code:
4 - European Region
Time Code:
Day Time
Night Time
Transaction Summary:
Duration of Call: 89 minute(s)
Destination Code: 4 - European Region
Time Code: Day Time
Total Charge: Php 1038.33
Compute Charge
Reset
About
Close


Call Charge Calculator
User Inputs:
Length of Call (in minutes):
Destination Code:
About
Hello I'm Your Call Charge Calculator
OK
Compute Charge
Reset
About
Close