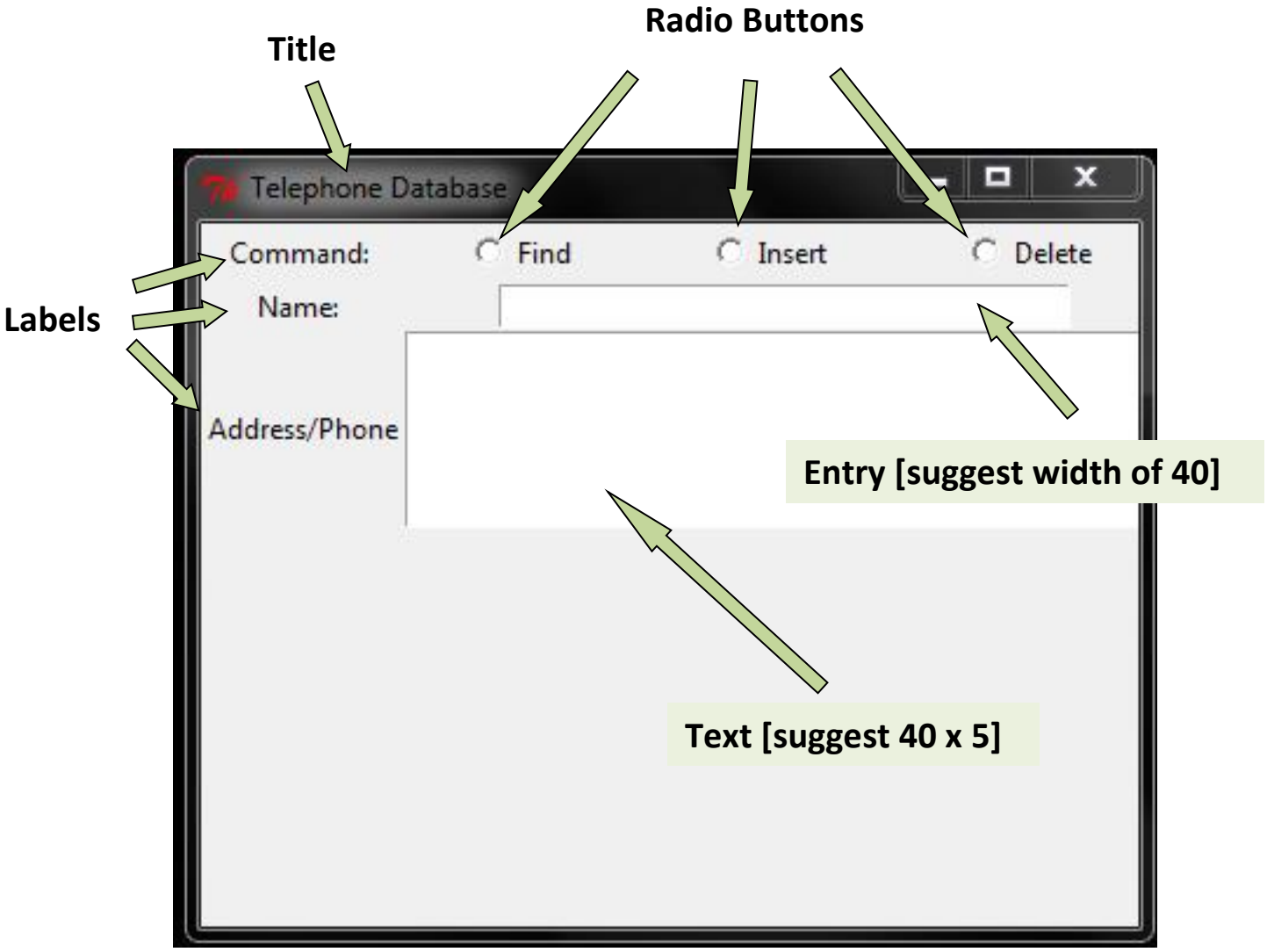
CSCI 1511 - Lab 10

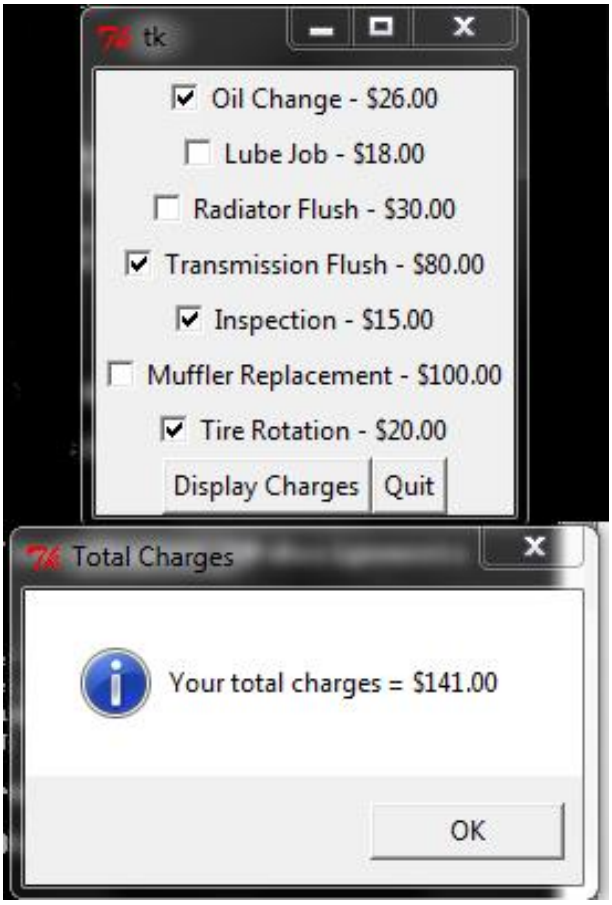
There are two projects you need to submit for Lab 10, #1 and your choice for the second project.

1. Submit the python code created for the “Mad Lib” program developed throughout chapter 10.
2. Telephone Database using *Tkinter*: Create an interface for a telephone database. The interface should look like the image below.
    1. The application should have size “400x200”.
    2. An *Entry* widget next to the label “Name:” for user input.
    3. A *Text* widget next to the label “Address/Phone” for displaying information
    4. The *Radio* buttons should be bound to three different functions, so when you click
        1. “Find”, you get the output “I’m trying to find that person.”
        2. “Insert”, you get the output “Putting the name in the database.”
        3. “Delete”, you get the output “Deleting entry.”
    5. The image below points out the necessary features.



NOTE: During the last weeks of this semester, we will rig the interface, using pickling, to execute the radio button functions.

3. Joe's Automotive Repair Shop.
   a. Performs the following routine maintenance services
      - Oil Change - $26.00
      - Lube Job - $18.00
      - Radiator Flush - $30.00
      - Transmission Flush - $80.00
      - Inspection - $15.00
      - Brake Rotors - $120.00
      - Muffler Replacement - $100.00
      - Tire Rotation - $20.00
   b. Write a GUI program with check buttons that allow the user to select a combination of the services list above.
   c. When the user is ready, they may click a button to display the total charges.
   d. Add a 'quit' button.



Each file should be a python program, a text file, extended by ".py". The header of each file should be comments which tell the name of the program, your name, and the date.

The box I used to show the total is called a messagebox. It is a Tkinter utility and is quite easy to use. You can include a single command in your calculation function when you wish to display the total. Here is an example taken from a web site. Google “tkinter messagebox”.

Example:

```
import tkinter
from tkinter import messagebox
from tkinter import *

class testMB():
    def __init__(self):
        # Create the main window
        self.main_window = tkinter.Tk()

        # Create frames
        self.top_frame = tkinter.Frame(self.main_window)
        self.bottom_frame = tkinter.Frame(self.main_window)
        self.image_frame = tkinter.Frame(self.main_window)
        can = tkinter.Canvas(self.image_frame, width = 200, height = 200)
        can.pack()
        img = tkinter.PhotoImage(file = "earth.gif")
        can.create_image((100, 100), image = img
        # Create the two buttons in the bottom frame
        self.display_button = tkinter.Button(self.top_frame,\
                              text = 'Be nice! Say hello!',\
                              command = self.hello)
        self.quit_button = tkinter.Button(self.bottom_frame,\
                              text = 'Quit',\
                              command =self.main_window.destroy)
        # Pack the widgets in the bottom frame
        self.display_button.pack(side='left')
        self.quit_button.pack(side='left')

        # Pack the frames
        self.image_frame.pack()
        self.top_frame.pack()
        self.bottom_frame.pack()

        # Enter the tkinter main loop
        self.main_window.mainloop()

    # Define the show_info function
    def hello(self):

        # Display message box
        tkinter.messagebox.showinfo(title = "Stubborn", message = "No!!!")

# Main
# Create an instance of AutoGUI
exampleMB = testMB()
```

When the above code is executed, it produces following result: