CS 162 Final project: Contact GUI

1. An example of global variables used and a simple for loop to format and display the first/ last name as well as the phone number.

```
def main_window():
    global fname, lname, phonevar, select                    #global variables
    win = Tk()
```

```
  for fname, lname, phone in phonelist:
      select.insert(END, "{0}, {1}, {2}".format(lname, fname, phone))
```

2. This is an example using the ".format" method. There are many other identifiers throughout the code.

```
def contact_display():
    phonelist.sort(key=lambda record: record[1])
    select.delete(0, END)
    for fname, lname, phone in phonelist:
        select.insert(END, "{0}, {1}, {2}".format(lname, fname, phone))
```

3. These are the main functions that power this program, there are also many methods used in the main file.

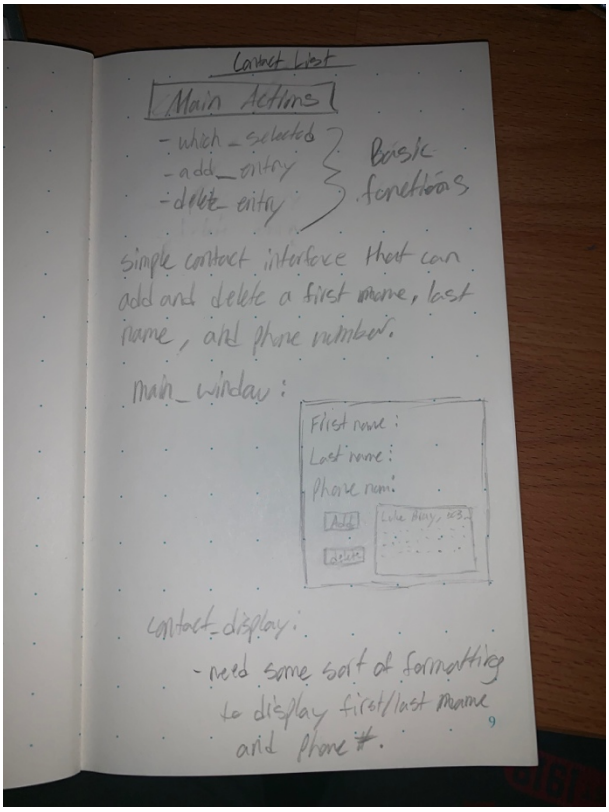
```
  def which_selected():
      print("At {0}".format(select.curselection()))
      return int(select.curselection()[0])

  def add_entry():
      phonelist.append([fname.get(), lname.get(), phonevar.get()])
      contact_display()

  def delete_entry():
      del phonelist[which_selected()]
      contact_display()

  def main_window():
      global fname, lname, phonevar, select
      win = Tk()
```

4. Here was my initial design build before starting this project. When I started writing the code for this project I thought it would be best to write out the main functions being used to get an understanding of everything that would be needed.



5. My best testing attempt… I'm still in the learning process of writing good tests and that's one of my main goals this summer.

```
test_add.py > ...
1   import unittest
2   from app_main import contact_display
3
4   # I'm not sure how to test this function in this situation.
5
6   class test_contact_display(unittest.Testcase):
7       def test_contact_display(self):
8           result = contact_display
9           self.assert result in (lname, fname, phone)
```

6. This piece initializes the text boxes that allows user input to take place, there's a few other important pieces of code that interact with this but this is the main part.

```
Label(frame1, text="First Name").grid(row=0, column=0, sticky=W)
fname = StringVar()
fname = Entry(frame1, textvariable=fname)
fname.grid(row=0, column=1, sticky=W)

Label(frame1, text="Last Name").grid(row=1, column=0, sticky=W)
lname = StringVar()
lname = Entry(frame1, textvariable=lname)
lname.grid(row=1, column=1, sticky=W)

Label(frame1, text="Phone").grid(row=2, column=0, sticky=W)
phonevar = StringVar()
phone = Entry(frame1, textvariable=phonevar)
phone.grid(row=2, column=1, sticky=W)
```

8. An example of the components which make up the GUI. I kind of used the event driven programming paradigm but that's another thing I need to get a better understanding of, especially in the design phase of a project.

```
def main_window():
    global fname, lname, phonevar, select                          #global variables
    win = Tk()

    frame1 = Frame(win)
    frame1.pack()

    Label(frame1, text="First Name").grid(row=0, column=0, sticky=W)
    fname = StringVar()
    fname = Entry(frame1, textvariable=fname)
    fname.grid(row=0, column=1, sticky=W)

    Label(frame1, text="Last Name").grid(row=1, column=0, sticky=W)
    lname = StringVar()
    lname = Entry(frame1, textvariable=lname)
    lname.grid(row=1, column=1, sticky=W)

    Label(frame1, text="Phone").grid(row=2, column=0, sticky=W)
    phonevar = StringVar()
    phone = Entry(frame1, textvariable=phonevar)
    phone.grid(row=2, column=1, sticky=W)

    frame2 = Frame(win)                                # Clickable buttons
    frame2.pack()
    b1 = Button(frame2, text = "Add", command=add_entry)
    b2 = Button(frame2, text = "Delete", command=delete_entry)
    b1.pack(side=LEFT)
    b2.pack(side=LEFT)

    frame3 = Frame(win)                               # list of names
    frame3.pack()
    scroll = Scrollbar(frame3, orient=VERTICAL)
    select = Listbox(frame3, yscrollcommand=scroll.set, height=6)
    scroll.config(command=select.yview)
    scroll.pack(side=RIGHT, fill=Y)
    select.pack(side=LEFT, fill=BOTH, expand=1)
    return win
```