class Purchase:  
 def \_\_init\_\_(self): # constructor  
 self.window = Tk() # tkinter root  
 self.window.title = ('Cash Register')  
 self.window.geometry("940x640")  
 self.window.resizable(False, False)  
 self.window.configure(bg="cornflowerblue")  
 self.counter = int(0) # counter var keeps count of how many items  
 self.itemNums = [] # 4 Arrays Item Nr, name, price and items in the cart  
 self.itemNames = []  
 self.itemPrices = []  
 self.presentItems = []  
 self.totalPrice = 0

Code above shows the class Purchase with the constructor. Then we open a window with a tkinter. We set size of the window declare variable counter that will be used later for counting the items in the cart.

Then, we declare four arrays. Each for name, price , item number and the present item that is in the cart.

def addItem(self): # Function is event handler / add item button  
 if(self.txtEntry.get() != ""): # If text field is not empty  
 self.presentItems.append(self.txtEntry.get()) # Add text to present items array  
 self.counter+=1 # increase item count by 1  
 self.listbox.insert(self.counter, str(self.counter) + ". Item Number: " + str(self.presentItems[-1]) +  
 " | Item Name: " + str(self.itemNames[int(self.presentItems[-1])]) + " | Item Price: $" + str(self.itemPrices[int(self.presentItems[-1])]))  
 self.txtEntry.delete(0, 'end')  
 self.totalPrice +=float(self.itemPrices[int(self.presentItems[-1])])  
 else:  
 messagebox.showinfo("Cash Register", "Textfield cannot be empty!") #warn the user if text field is empty

In the code above the additem function is shown. This is the event handler function for the add item button. In this function, we grab the text from the text field which is the item number. Then we match it with the other arrays that contain the other data. We display the data on the listbox on the right side.

def doTotal(self): #this function calculates the total  
 self.counter +=1  
 total = float(round(self.totalPrice, 2))  
 self.listbox.insert(self.counter,"Total: $" + str(total))

The code above shows the doTotal function which calculates the total in the cart.

def layout(self): #this function is the UI layout  
 lbl01 = Label(self.window, text='Insert Item Number Below', font=("Arial", 18))  
 lbl01.config(bg='lightblue1')  
 self.txtEntry = Entry(self.window, width=50)  
 self.btnAdd = Button(self.window, text="Add Item", fg="red", command=self.addItem, font="Verdana 18", bd=2, bg="lightblue",  
 relief="groove")  
 self.btnTotal = Button(self.window, text="Calculate Total", fg="red", command=self.doTotal, font="Verdana 18", bd=2,  
 bg="lightblue",relief="groove")  
 self.listbox = Listbox(self.window, height=35, width=80, bd=2)  
 lbl01.grid(row=0,column=0,sticky=NE, ipadx=10, ipady=50, pady=5, padx=5)  
 self.txtEntry.grid(row=1, column=0, sticky=NE, ipady=10, pady=5, padx=5)  
 self.btnAdd.grid(row=2, column=0, ipady=0)  
 self.btnTotal.grid(row=3, column=0, ipady=0)  
 self.listbox.grid(row=0, column=2, rowspan=4, sticky=W, padx=75, pady=10)

This is the layout function which provides the UI for the application. There are one label, one textfield entry, two buttons, and listbox. All these widgets are sitting inside of a grid of four rows and two columns. The list box spans in the second column over four rows. The buttons have two event handler functions, additem, dototal respectively.

def addToArrays(self): #this function takes files and adds them into three separate arrays  
 with open("itemNumsFile.txt") as f1:  
 self.itemNums = f1.readlines()  
 with open("itemNamesFile.txt") as f2:  
 self.itemNames = f2.readlines()  
 with open("itemPricesFile.txt") as f3:  
 self.itemPrices = f3.readlines()

This function above takes data from the text files and adds to three arrays. Item number, name, price.

def loop(self): # tkinter loop  
 self.window.mainloop()

Above function loop runs the tkinter loop.