

Lab Exercises - 3 Solutions

Files

Exercise 1 – Display the following menu to the user:

- 1) Create a new file
- 2) Display the file
- 3) Add a new item to the file

Make a selection to 1, 2 or 3

Ask the user to enter 1, 2 or 3. If they select anything other than 1, 2 and 3, it should display a suitable error message.

If they select 1, ask the user to enter a school subject and save it to a new file called "Subjects.txt". It should overwrite any existing file with a new file. If they select 2, display the contents of "Subjects.txt" file. If they select 3, ask the user to enter a new subject and save it to the file and then display the entire content of the file.

Run the programme several times to test the options.

```
print ("1) Create a new file")
print ("2) Display the file")
print("3) Add a new item to the file")
selection = int(input("Make a selection 1, 2 or 3: "))
if selection == 1:
    subject = input("Enter a school subject: ")
    file = open("Subject.txt", "w")
    file.write(subject + "\n")
    file.close()
elif selection == 2:
    file = open("Subject.txt", "r")
    print(file.read())
elif selection == 3:
    file = open("Subject.txt", "a")
    subject = input("Enter a school subject: ")
    file.write(subject + "\n")
    file.close()
    file = open("Subject.txt", "r")
    print(file.read())
else:
    print("Invalid option")
```

Exercise 2 - Create a simple maths quiz that will ask the user for their name and then generate two random questions. Store their name, the questions that were asked, their answers and their final score in a .csv file. Whenever the programme is run it should add to the csv file, and not overwrite anything.

```

1 import csv
2 import random
3
4 score = 0
5 name = input("What is your name: ")
6 q1_num1 = random.randint(10,50)
7 q1_num2 = random.randint(10,50)
8 question1 = str(q1_num1) + "+" + str(q1_num2) + " = "
9 ans1 = int(input(question1))
10 realans1 = q1_num1 + q1_num2
11 if ans1 == realans1:
12     score = score + 1
13 q2_num1 = random.randint(10,50)
14 q2_num2 = random.randint(10,50)
15 question2 = str(q1_num1) + " + " + str(q2_num2) + " = "
16 ans2 = int(input(question2))
17 realans2 = q2_num1 + q2_num2
18 if ans2 == realans2:
19     score = score + 1
20
21 file = open("QuizScore.csv", "a")
22 newrecord = name+", "+question1+", "+str(ans1)+", "+question2+", "+str(ans2)+", "+str(score)+"\n"
23 file.write(str(newrecord))

```

Exercise 3 - (Function and .csv files) Create the following menu:

- 1) Add to file
- 2) View all records
- 3) Quit programme

Enter the number of your selection:

If the user selects 1, allow them to add to a file called Salaries.csv which will store their name and salary. If they select 2 it should display all records in the Salaries.csv file. If they select 3 it should stop the programme. If they select an incorrect option, they should see an error message. They should keep returning to the menu until they select option 3.

```

import csv

def addtofile():
    file = open("Salaries.csv", "a")
    name = input("Enter name: ")
    salary = int(input("Enter Salary: "))
    newrecord = name + ", " + str(salary) + "\n"
    file.write(str(newrecord))
    file.close()

def viewrecords():
    file = open("salaries.csv", "r")
    for row in file:
        print(row)
    file.close()

tryagain = True
while tryagain == True:
    print("1) Add to file")
    print("2) View all records")
    print("3) Quit program")
    print()
    selection = input("Enter the number of your selection: ")
    if selection == "1":
        addtofile()
    elif selection == "2":
        viewrecords()
    elif selection == "3":
        tryagain = False
    else:
        print("Incorrect option")

```

Exercise 4 – (function and csv) In Python, it is not technically possible to directly delete a record from a .csv file. Instead, you need to save the file to a temporary list in Python, make the changes to the list and then overwrite the original file with the temporary list.

Change the previous programme to allow you to do this. Your menu should now look like this:

- 1) Add to file
- 2) View all records
- 3) Delete a record
- 4) Quit programme

```

import csv

def addtofile():
    file = open("Salaries.csv", "a")
    name = input("Enter name: ")
    salary = int(input("Enter Salary: "))
    newrecord = name + ", " + str(salary) + "\n"
    file.write(str(newrecord))
    file.close()

def viewrecords():
    file = open("salaries.csv", "r")
    for row in file:
        print(row)
    file.close()

def deleterecord():
    file = open("Salaries.csv", "r")
    x = 0
    tmplist = []
    for row in file:
        tmplist.append(row)
    file.close()
    for row in tmplist:
        print(x, row)
        x = x + 1
    rowtodelete = int(input("Enter thr row number to delete: "))
    del tmplist[rowtodelete]
    file = open("Salaries.csv", "w")
    for row in tmplist:
        file.write(row)
    file.close()

tryagain = True
while tryagain == True:
    print("1) Add to file")
    print("2) View all records")
    print("3) Delete a record")
    print("4) Quit programme")
    print()
    selection = input("Enter the number of your selection: ")
    if selection == "1":
        addtofile()
    elif selection == "2":
        viewrecords()
    elif selection == "3":
        deleterecord()
    elif selection == "4":
        tryagain = False
    else:
        print("Incorrect option")

```

Exercise 5 - Sum a Collection of Numbers

Create a program that sums all of the numbers entered by the user while ignoring any input that is not a valid number. Your program should display the current sum after each number is entered. It should display an appropriate message after each non-numeric input, and then continue to sum any additional numbers entered by the user. Exit the program when the user enters a blank line. Ensures that your program works correctly for both integer and floating-point numbers.

```
# Read the first line of input from the user
line = input("Enter a number: ")
total = 0

# Keep reading until the user enters a blank line
while line != "":
    try:
        # Try and convert the line to a number
        num = float(line)
        # If the conversion succeeds then add it to the total and display it
        total = total + num
        print("The total is now", total)

    except ValueError:
        # Display an error message before going on to read the next value
        print("That wasn't a number.")

    line = input("Enter a number: ")

# Display the total
print("The grand total is", total)
```

Tkinter GUI

Exercise 6 – Write a programme that can be used instead of rolling a six-sided die in a board game. When a user clicks a button, it should display a random whole number between 1 and 6 (inclusive).

```
from tkinter import *
import random

def click():
    num = random.randint(1,6)
    answer["text"] = num

window = Tk()
window.title("Roll a dice")
window.geometry("100x120")

button1 = Button(text = "Roll", command = click)
button1.place(x = 30, y = 30, width = 50, height = 25)

answer = Message(text = "")
answer.place(x = 40, y = 70, width = 30, height = 25)

window.mainloop()
```

Exercise 7 – Create a programme that will ask the user to enter a number in a box. When they click on a button it will add that number to a total and display it in another box. This can be repeated as many times as they want and keep adding to the total. There should be another button that resets the total back to 0 and empties the original text box, ready for them again.

```
from tkinter import *

def add_on():
    num = enter_txt.get()
    num = int(num)
    answer = output_txt["text"]
    answer = int(answer)
    total = num + answer
    output_txt["text"] = total

def reset():
    total = 0
    output_txt["text"] = 0
    enter_txt.delete(0, END)
    enter_txt.focus()

total = 0
num = 0

window = Tk()
window.title("Adding Together")
window.geometry("450x100")

enter_lbl = Label(text = "Enter a number:")
enter_lbl.place(x = 50, y = 20, width = 100, height = 25)

enter_txt = Entry(text = 0)
enter_txt.place(x = 150, y = 20, width = 100, height = 25)
enter_txt["justify"] = "center"
enter_txt.focus()

add_btn = Button(text = "Add", command = add_on)
add_btn.place(x = 300, y = 20, width = 50, height = 25)

output_lbl = Label(text = "Answer = ")
output_lbl.place(x = 50, y = 50, width = 100, height = 25)

output_txt = Message(text = 0)
output_txt.place(x = 150, y = 50, width = 100, height = 25)
output_txt["bg"] = "white"
output_txt["relief"] = "sunken"

clear_btn = Button(text = "Clear", command = reset)
clear_btn.place(x = 300, y = 50, width = 50, height = 25)

window.mainloop()
```

Exercise 8 – Create a window that will ask the user to enter a name in a text box. When they click on a button, it will add it to the end of the list that is displayed on the screen. Create another button that will clear the list.

```
from tkinter import *
def add_name():
    name = name_box.get()
    name_list.insert(END, name)
    name_box.delete(0, END)
    name_box.focus()

def clear_list():
    name_list.delete(0, END)
    name_box.focus()

window = Tk()
window.title("Names list")
window.geometry("400x200")

label1 = Label(text = "Enter a name: ")
label1.place(x = 20, y = 20, width = 100, height = 25)

name_box = Entry(text = "")
name_box.place(x = 120, y = 20, width = 100, height = 25)
name_box.focus()

button1 = Button(text = "Add to list", command = add_name)
button1.place(x = 250, y = 20, width = 100, height = 25)

name_list = Listbox()
name_list.place(x = 120, y = 50, width = 100, height = 100)

button2 = Button(text = "Clear list", command = clear_list)
button2.place(x = 250, y = 50, width = 100, height = 25)

window.mainloop()
```

Exercise 9 – Create a window that will ask the user to enter a number in a text box. When they click on a button, it will use the code `variable.isdigit()` to check if it is a whole number. If it is a whole number, add it to a list box, otherwise clear the entry box. Add another button that will clear the list. Alter the programme 129 to add a third button that will save the list to a .csv file. The code `tmp_list = num_list.get(0, END)` can be used to save the contents of a list box as a tuple called `tmp_list`.

```

3 from tkinter import *
4 import csv
5
6 def add_number():
7     num = num_box.get()
8     if num.isdigit():
9         num_list.insert(END,num)
10        num_box.delete(0, END)
11        num_box.focus()
12    else:
13        num_box.delete(0, END)
14        num_box.focus()
15
16 def clear_list():
17     num_list.delete(0, END)
18     num_box.focus()
19
20 def save_list():
21     file = open("numbers.csv", "w")
22     tmp_list=num_list.get(0, END)
23     item = 0
24     for x in tmp_list:
25         newrecord = tmp_list[item] + "\n"
26         item = item + 1
27     file.close()
28
29 window = Tk()
30 window.title("Number list")
31 window.geometry("400x200")
32
33 label1 = Label(text = "Enter a number: ")
34 label1.place(x = 20, y = 20, width = 100, height = 25)
35
36 num_box = Entry(text = 0)
37 num_box.place(x = 120, y = 20, width = 100, height = 25)
38 num_box.focus()
39
40 button1 = Button(text = "Add to list", command = add_number)
41 button1.place(x = 250, y = 20, width = 100, height = 25)
42
43 num_list = Listbox()
44 num_list.place(x = 120, y = 50, width = 100, height = 100)
45
46 button2 = Button(text = "Clear list", command = clear_list)
47 button2.place(x = 250, y = 50, width = 100, height = 25)
48
49 button3 = Button(text = "Save list", command = clear_list)
50 button3.place(x = 250, y = 80, width = 100, height = 25)
51
52 window.mainloop()

```


Exercise 10 – Create a programme that will allow the user to create a new .csv file. It should ask them to enter the name and age of a person and then allow them to add this to the end of the file they have just created.

```
8 from tkinter import *
9 import csv
10
11 def create_new():
12     file = open("ages.csv", "w")
13     file.close()
14
15 def save_list():
16     file = open("ages.csv", "a")
17     name = name_box.get()
18     age = age_box.get()
19     newrecord = name + "," + age + "\n"
20     file.write(str(newrecord))
21     file.close()
22     name_box.delete(0, END)
23     age_box.delete(0, END)
24     name_box.focus()
25
26 window = Tk()
27 window.title("People List")
28 window.geometry("400x100")
29
30 label1 = Label(text = "Enter a name: ")
31 label1.place(x = 20, y = 20, width = 100, height = 25)
32
33 name_box = Entry(text = "")
34 name_box.place(x = 120, y = 20, width = 100, height = 25)
35 name_box["justify"] = "left"
36 name_box.focus()
37
38 label2 = Label(text = "Enter their age:")
39 label2.place(x = 20, y = 50, width = 100, height = 25)
40
41 age_box = Entry(text = "")
42 age_box.place(x = 120, y = 50, width = 100, height = 25)
43 age_box["justify"] = "left"
44
45 button1 = Button(text = "Create new file", command = create_new)
46 button1.place(x = 250, y = 20, width = 100, height = 25)
47
48 button2 = Button(text = "Add to file", command = save_list)
49 button2.place(x = 250, y = 50, width = 100, height = 25)
50
51 window.mainloop()
```

Exercise 11 – Using the .csv file you created in 10, create a programme that will allow people to add names and ages to the list and create a button that will display the contents of the .csv file by importing it to a list box.

Exercise 12 – Create your own icon. Create a logo which measures 200 x 150, using Paint or another graphics package. Create a window like this using your icon and your logo



When the user enters their name and clicks on the Press Me button, it should display “Hello” and their name in the second text box.

Exercise 13 – Create a new programme that will generate two random whole numbers between 10 and 50. It should ask the user to add the numbers together and type in the answer. If they get the answer correct, display a suitable image such as a tick; if they get the answer wrong display a cross. They should click on a Next button to get another question.

Exercise 14 – Create a simple programme that shows a drop-down list containing several colours and a Click Me button. When the user selects a colour from the list and clicks the button it should change the background of the window to that colour. For an extra challenge, try to avoid using and if statement to do this.

Exercise 15 – Create a programme that will ask the user to enter a name and then select the gender for that person from the drop-down list. It should then add the name and the gender (separated by a comma) to a list box when the user clicks on a button.

Exercise 16 – Change the previous programme -15 so that when a new name and gender is added to the list box, it is also written to a text file. Add another button that will display the entire text file in the main Python shell window.

SQLite3

Exercise 17 – Create an SQL database called PhoneBook1 that contains a table called Names with the following data:

ID	First Name	Surname	Phone Number
1	Simon	Pierre	0142678 9056
2	Katarina	Iglesias	0203456 7078
3	Derrick	Brown	0122345 8765
4	John	Smith	0112653 2312
5	Mark	Isaac	01416571383

```

import sqlite3
# Connect to the database called PhoneBook or create one if there is none
with sqlite3.connect("PhoneBook1.db") as db:
    cursor = db.cursor()

    # Create a table called Names with four fields
    cursor.execute(""" CREATE TABLE IF NOT EXISTS Names(
id integer PRIMARY KEY,
firstname text,
surname text,
phonenumber text); """)

# Insert data into the table
cursor.execute(""" INSERT INTO Names(id,firstname,surname,phonenumber)
VALUES ("60", "Simon","Pierre","0142678 9056")""")
db.commit() # Saves the changes

# Insert data into the table Names
cursor.execute(""" INSERT INTO Names(id, firstname,surname,phonenumber)
VALUES ("70", "Katarina","Iglesias","0203456 7078")""")
db.commit() # saves the changes

# Insert data into a table called Names
cursor.execute(""" INSERT INTO Names(id,firstname,surname,phonenumber)
VALUES ("30", "Derrick", "Brown", "0122345 8765")""")
db.commit() # saves the changes

# Insert data into a table called Names
cursor.execute(""" INSERT INTO Names(id,firstname,surname,phonenumber)
VALUES ("40", "John", "Smith", "0112653 2312")""")
db.commit() # saves the changes

# Insert data into a table called Names
cursor.execute(""" INSERT INTO Names(id,firstname,surname,phonenumber)
VALUES ("50", "Mark", "Isaac", "0141657 1383")""")
db.commit() # saves the changes

db.close() # close the database

```

Exercise 18 – Using the phonebook database, write a programme that will display the following menu

Main menu

- 1) View phone book
 - 2) Add to phone book
 - 3) Search for surname
 - 4) Delete person from phone book
 - 5) Quit
- Enter your selection

If the user selects 1, they should be able to view the entire phonebook. If they select 2, it should allow them to add a new person to the phonebook. If they select 3, it should ask them for a surname and then display only the record of people with the same surname. If they select 4, it should ask for

an ID and then delete that record from the table. If they select 5, it should end the programme. Finally, the programme should display a suitable message if they enter an incorrect selection from the menu. They should return to the menu after each action, until they select 5.

```
def viewphonebook():
    cursor.execute("SELECT * FROM Names")
    for x in cursor.fetchall():
        print(x)
def addtophonebook():
    newid = int(input("Enter ID: "))
    newfname = input("Enter first name: ")
    newsname = input("Enter surname: ")
    newpnum = input("Enter phone number: ")
    cursor.execute("""INSERT INTO Names (id,firstname,surname,phonenumber)
        VALUES (?,?,,?)""", (newid,newfname,newsname,newpnum))
    db.commit()

def selectname():
    selectsurname = input("Enter a surname: ")
    cursor.execute("SELECT * FROM Names WHERE surname = ?", [selectsurname])
    for x in cursor.fetchall():
        print(x)

def deletedata():
    selectid = int(input("Enter ID: "))
    cursor.execute("DELETE FROM Names WHERE id = ?", [selectid])
    cursor.execute("SELECT * FROM Names")
    for x in cursor.fetchall():
        print(x)
    db.commit()

with sqlite3.connect("PhoneBook1.db") as db:
    cursor = db.cursor()

def main():
    again = "y"
    while again == "y":
        print()
        print("Main Menu")
        print()
        print("1) View phone book")
        print("2) Add to phone book")
        print("3) Search for surname")
        print("4) Delete person from phone book")
        print("5) Quit")
        print()
        selection = int(input("Enter your selection: "))
        print()

        if selection == 1:
            viewphonebook()
        elif selection == 2:
            addtophonebook()
        elif selection == 3:
            selectname()
        elif selection == 4:
            deletedata()
        elif selection == 5:
            again = "n"
        else:
            print("Incorrect selection entered")

main()
db.close()
```