Team Application Exercises (tAPP-3)

Problem 1

Write a code that creates a database called employees. That database contains a table called contactDetails. The following output will be produced when the program is executed.

```
(42, 'Derrick', 'Brown', '0122345 8765')
(62, 'Simon', 'Pierre', '0142678 9056')
(72, 'Katarina', 'Iglesias', '0203456 7078')
```

Problem 1 Solution

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

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

# Insert data into the table
cursor.execute(""" INSERT INTO contactDetails(id,firstname,surname,phonenumber)
VALUES ("42", "Simon","Pierre","0142678 9056")""")
db.commit() # Saves the changes

# Insert data into the table Names
cursor.execute(""" INSERT INTO contactDetails(id, firstname,surname,phonenumber)
VALUES ("62", "Katarina","Iglesias","0203456 7078")""")
db.commit() # saves the chnages

# Insert data into a table called Names
cursor.execute(""" INSERT INTO contactDetails(id,firstname,surname,phonenumber)
VALUES ("72", "Derrick", "Brown", "0122345 8765")""")
db.commit() # saves the changes


cursor.execute("SELECT * FROM contactDetails")
for x in cursor.fetchall():
    print(x)

db.close() # close the database
```

Problem 2 - Debugging code

This code should 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. However, it does not execute due to some errors.

Task: Identify the errors (making sure you indicate the line number) and how many errors have you found?

Problem 2 – Solution

There are 9 errors:

Line 12: wrong file extension ages.cvs, it should be csv

Line 16: wrong mode - It should be append "a". 'x' mode creates a new file and writes to that file. If the file already exits, the programme will crash rather than overwriting it. You do not want the programme to crash.

Line 19: comma missing between the quotes. The comma could be omitted if you do not want it in the output. However, we are saving into a CSV file and we want to save the name and age in different columns. And a backslash is missing before "n". without it will save everything in the same row.

Line 20 it should be str(newrecord) instead of srt(newrecord).

Line 35: left square bracket is missing.

Line 42: There is an extra parenthesis.

Line 46: a dot is missing between button1 and place

Line 48: Quotes are missing around string "Add to file"

Line 51: It should be window.mainloop()

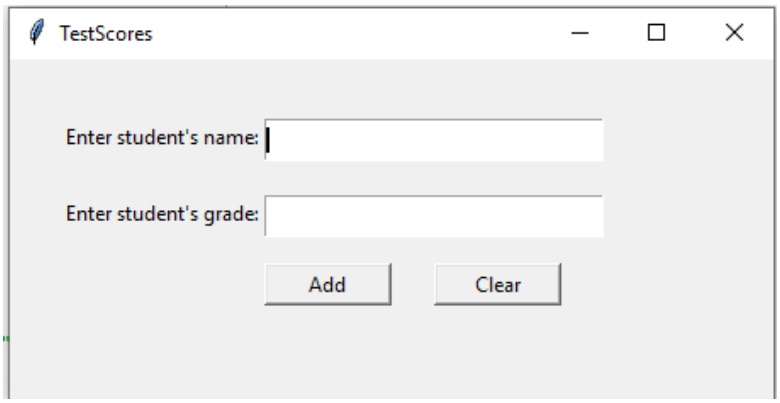
```
8 from tkinter import *
9
10
11 def create_new():
12     file = open("ages.cvs", "w")
13     file.close()
14
15 def save_list():
16     file = open("ages.csv", "x")
17     name = name_box.get()
18     age = age_box.get()
19     newrecord = name + " " + age + "n"
20     file.write(srt(newrecord))
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.loop()
```

Correct code

```
from tkinter import *
import csv

def create_new():
    file = open("ages.csv", "w")
    file.close()

def save_list():
    file = open("ages.csv", "a")
    name = name_box.get()
    age = age_box.get()
    newrecord = name + "," + age + "\n"
    file.write(str(newrecord))
    file.close()
    name_box.delete(0, END)
    age_box.delete(0, END)
    name_box.focus()

window = Tk()
window.title("People List")
window.geometry("400x100")

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["justify"] = "left"
name_box.focus()

label2 = Label(text = "Enter their age:")
label2.place(x = 20, y = 50, width = 100, height = 25)

age_box = Entry(text = "")
age_box.place(x = 120, y = 50, width = 100, height = 25)
age_box["justify"] = "left"

button1 = Button(text = "Create new file", command = create_new)
button1.place(x = 250, y = 20, width = 100, height = 25)

button2 = Button(text = "Add to file", command = save_list)
button2.place(x = 250, y = 50, width = 100, height = 25)

window.mainloop()
```

Problem 3 - Parsons puzzle

When this program code runs, it displays the following screen:



It saves the data to an SQL database called TestScores when the Add button is clicked. The clear Button clears the Window.

Task:

Below are the lines of code for a solution. However, they are not in the correct order. Rearrange the code by entering the number that corresponds to each statement/block of statements (the first number has been entered for you).

Problem 3 Solution

Number	Code line/block
17	`sname = Entry(text = "")`
14	`window.geometry("450x200")`

7	`db.commit()`
	`sname.delete(0, END)`
	`sgrade.delete(0, END)`
	`sname.focus()`

15	`label1 = Label(text = "Enter student's name:")`
26	`clearbtn.place(x = 250, y = 120, width = 75, height = 25)`
4	`newname = sname.get()`
19	`label2 = Label(text = "Enter student's grade:")`
11	`cursor.execute(""" CREATE TABLE IF NOT EXISTS Scores(` `id integer PRIMARY KEY, name text, score integer); """)`
3	`def addtolist():`
2	`from tkinter import *`
28	`db.close()`
21	`sgrade = Entry(text = "")`
25	`clearbtn = Button(text = "Clear", command = clearlist)`
12	`window = Tk()`
5	`newgrade = sgrade.get()`
24	`addbtn.place(x = 150, y = 120, width = 75, height = 25)`
18	`sname.place(x = 150, y = 35, width = 200, height = 25)` `sname.focus()`
16	`label1.place(x = 30, y = 35)`
23	`addbtn = Button(text = "Add", command = addtolist)`
6	`cursor.execute(""" INSERT INTO Scores(name,score)` `VALUES (?,?)""",(newname,newgrade))`
20	`label2.place(x = 30, y = 80)`
1	`import sqlite3`
13	`window.title("TestScores")`
8	`def clearlist():`
22	`sgrade.place(x = 150, y = 80, width = 200, height = 25)` `sgrade.focus()`
10	`with sqlite3.connect("TestScore.db") as db:` `cursor = db.cursor()`
27	`window.mainloop()`

9	`sname.delete(0, END)`
	`sgrade.delete(0, END)`
	`sname.focus()`

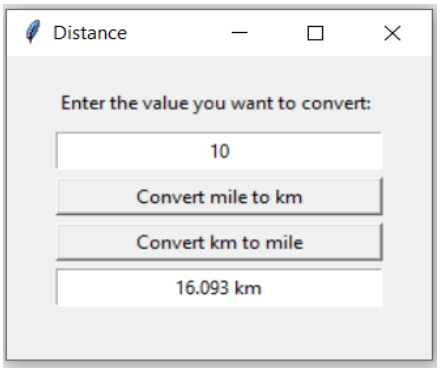
Complete code:

```
import sqlite3
from tkinter import *

def addtolist():
    newname = sname.get()
    newgrade = sgrade.get()
    cursor.execute(""" INSERT INTO Scores(name,score)
 VALUES (?,?)""",(newname,newgrade))
    db.commit()
    sname.delete(0, END)
    sgrade.delete(0, END)
    sname.focus()

def clearlist():
    sname.delete(0, END)
    sgrade.delete(0, END)
    sname.focus()

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

cursor.execute(""" CREATE TABLE IF NOT EXISTS Scores(
        id integer PRIMARY KEY, name text, score integer); """)

window = Tk()
window.title("TestScores")
window.geometry("450x200")

label1 = Label(text = "Enter student's name:")
label1.place(x = 30, y = 35)
sname = Entry(text = "")
sname.place(x = 150, y = 35, width = 200, height = 25)
sname.focus()
label2 = Label(text = "Enter student's grade:")
label2.place(x = 30, y = 80)
sgrade = Entry(text = "")
sgrade.place(x = 150, y = 80, width = 200, height = 25)
sgrade.focus()
addbtn = Button(text = "Add", command = addtolist)
addbtn.place(x = 150, y = 120, width = 75, height = 25)
clearbtn = Button(text = "Clear", command = clearlist)
clearbtn.place(x = 250, y = 120, width = 75, height = 25)

window.mainloop()
db.close()
```

Problem 4:

This program below should allow the user to convert between miles and kilometres. 1 kilometre = 0.6214 miles and 1 mile = 1.6093 kilometres.

Task: Complete the code (with correct indentation).




```
from tkinter import *
def convert1():
    mile = textbox1.get()
    mile = int(mile)
    message1 = mile * 1.6093
    textbox2.delete(0, END)
    textbox2.insert(END, message1)
    textbox2.insert(END, " km")

def convert2():
    km = textbox1.get()
    km = int(km)
    message = km * 0.6214
    textbox2.delete(0, END)
    textbox2.insert(END, message)
    textbox2.insert(END, " miles")

window = Tk()
window.title("Distance")
window.geometry("260x200")

label1 = Label(text = "Enter the value you want to convert: ")
label1.place(x = 30, y = 20)

textbox1 = Entry(text = "")
textbox1.place(x = 30, y = 50, width = 200, height = 25)
textbox1["justify"] = "center"
textbox1.focus()

convert1 = Button(text = "Convert mile to km", command = convert1)
convert1.place(x = 30, y = 80, width = 200, height = 25)

textbox2 = Entry(text = "")
textbox2.place(x = 30, y = 140, width = 200, height = 25)
textbox2["justify"] = "center"

convert2 = Button(text = "Convert km to mile", command = convert2)
convert2.place(x = 30, y = 110, width = 200, height = 25)
window.mainloop()
```