# Worksheet 6 Files and exception handling Answers

**Task 1: Processing a text file**

The incomplete pseudocode program given below is intended to store data from a recent one-day Maths workshop. The data, consisting of a name and a test mark for each student attending, will be input by the organiser and saved on a disk file.

 The program starts by displaying a menu of options:

 MENU

 1 Input data and save to new file

 2 Input data and append to existing file

 3 Calculate and display average mark

 4 Display data

 5 Quit program

 The program has a modular structure with the menu display and each of the first 4 options being written as a separate subroutine called from the main program.

 The basic program structure, written in pseudocode, is given below.

 Answer questions (a)-(f) given below the pseudocode.

#subroutine to display menu and validate choice

SUB displayMenu()

 #display menu of options

 OUTPUT("1. Input data and save to new file")

 OUTPUT("2. Input data and append to existing file")

 OUTPUT("3. Calculate and display average mark")

 OUTPUT("4. Display data")

 OUTPUT("5. Quit")

 choice = USERINPUT("Enter your choice: ")

 WHILE choice < "1" or choice > "5"

 choice = USERINPUT("Invalid choice - please re-enter: ")

 ENDWHILE

 RETURN choice

ENDSUB

#subroutine to input data and save to a new file

SUB saveToFile(openMode)

 DEFINE studentRecord as two fields: studentName, studentMark

 testResultsFile = OPEN("studentNamesfile.txt",openMode)

 studentMark = "0"

 studentName = INPUT ("Enter a student name, xxx to finish : ")

 WHILE studentName <> "xxx"

 studentMark = USERINPUT ("Enter mark : ")

 WRITE(testResultsFile, studentRecord)

 studentName = input ("Enter a student name, xxx to finish : ")

 ENDWHILE

 CLOSE testResultsFile

ENDSUB

#subroutine to calculate and display average mark

SUB calculateAverage()

 DEFINE studentRecord as two fields: studentName, studentMark

 testResultsFile = OPEN("studentNamesfile.txt",”read”)

 READ (testResultsFile, studentRecord)

 total = 0

 numRecs = 0

 WHILE NOT EOF

 total = total + studentMark

 numRecs = numRecs + 1

 READ (testResultsFile, studentRecord)

 END WHILE

 average = total/numRecs

 OUTPUT("Average mark:", average)

ENDSUB

#subroutine to display data

SUB displayData()

 DEFINE studentRecord as two fields: studentName, studentMark

 testResultsFile = OPEN("studentNamesfile.txt", “read”)

 READ (testResultsFile, studentRecord)

 WHILE NOT EOF

 OUTPUT(studentName,studentMark)

 READ (testResultsFile, studentRecord)

 ENDWHILE

ENDSUB

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main program starts here \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

option = displayMenu()

WHILE option <>"5"

 IF option = "1"

 saveToFile("write")

 ELSE IF option == "2"

 saveToFile("append")

 ELSE IF option == "3"

 name,mark = readIntoArray()

 calculateAverage(mark)

 ELSE IF option = "4"

 displayData()

 option = displayMenu()

ENDWHILE

OUTPUT("You have chosen to quit the program")

(a) Look at the main program. Explain what the following statement does:

 IF option = "1"

 saveToFile("write")

 If the user selects option 1, it calls the subroutine, passing it the parameter “write” so that file will be opened for writing to.

(b) In the subroutine **saveToFile**, what mode is the file opened in? Where is the file saved?

#subroutine to input data and save to a new file

SUB saveToFile(openMode)

 #open testResultFile in correct mode

 testResultsFile = OPEN("studentNamesfile.txt",openMode)

 It depends where it is called from. If menu option 1 is chosen, it is opened in **Write** mode.

 If menu option 2 is chosen it is opened in **Append** mode.

(c) Name three local variables in this subroutine. testResultsFile studentName, studentMark

(d) In the subroutines **calculateAverage** and **displayData**, what fields are contained in **studentRecord**?

 studentName and studentMark

(e) Complete the pseudocode program, inserting missing lines where indicated.

 (see red lines in pseudocode)

(f) From this pseudocode, write the program in a language of your choice.

 The Python program W6 Task 1 Maths workshop.py is in the Topic 6 Python programs folder.

 The Python program W6 Task 1 using array.py is an alternative version showing how to read the data into a list.

 The VB program W6 Task1 Maths Workshop.vb is in the Topic 6 VB programs folder.

**Task 2 : Binary files**

Look at the pseudocode given below.

 #set a buffer size so memory does not overflow

 #the graphic is 477KB

 buffersize 🡨 50000

 picturefile 🡨 OPEN('waterlilies.jpg','rb')

 newfilename 🡨 USERINPUT("Enter the new file name to copy image to: ")

 copiedfile 🡨 OPEN(newfilename,'wb') #open binary file for writing

 #read 50000 bytes at a time

 buffer 🡨 READ(pictureFile,buffersize)

 WHILE NOT end of file

 WRITE(copiedFile, buffer)

  **OUTPUT('.', end = ' ') #this outputs without moving to a new line**

 buffer 🡨 READ(pictureFile,buffersize)

 ENDWHILE

(a) What does the pseudocode do?

 It ask the user to enter a new file name and then copies a .jpg file to the new file

(b) If the .jpg file is 477KB, how many times will the WHILE loop be performed? 10 times

(c) what will be displayed by the OUTPUT statement shown in bold?

  **OUTPUT('.', end = ' ')**

 **. . . . . . . . . .** (i.e. 10 dots)

What is the purpose of this line?

 It reassures the programmer that something is going on, especially if the file to be copied is large!

(d) Where will the new file be located?

 In the same folder as the program.

See Python/VB programs **W6 Task 2 binary graphic file.py/vb**.

Students can write the program, selecting any image to copy. It needs to be moved into their current folder before running the program.

**Task 3 : Exception handling**

The following pseudocode records audience number on different nights for a theatre.

 DEFINE seatsRec as two fields, performanceDate and seatsSold

fileName = USERINPUT ("Please enter file name to record seat sales: ")

theatreFile = OPEN(fileName,"a")

performanceDate = input("Enter performance date, ddmmyy (xxx to end): ")

WHILE performanceDate <> "xxx":

 validSeatsSold = False

 WHILE NOT validSeatsSold

 seatsSold = input("Enter seats sold: ")

 TRY:

 integerSeats = int(seatsSold) #convert from text to integer

 validSeatsSold = True

 EXCEPT:

 print("Invalid entry ... please re-enter.")

 ENDWHILE

 WRITE(theatreFile, seatsRec)

 performanceDate = USERINPUT("Performance date, ddmmyy (xxx to end)")

ENDWHILE

theatreFile.close

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Part 2 of program \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#now reopen the file for reading

DEFINE seatsRec as two fields: performanceDate and seatsSold

OUTPUT("Reading audience numbers...")

fileName = USERINPUT ("Please enter file name to read seat sales: ")

TRY

 theatreFile = OPEN(fileName,"r")

 fileOpened = True

EXCEPT

 OUTPUT ("File not found ... program ending ")

 fileOpened = False

IF fileOpened

 READ (theatreFile, seatsRec)

 WHILE NOT end of file

 OUTPUT (“Date: ”,performanceDate, “Seats sold: ”, seatsSold)

 READ (theatreFile, seatsRec)

 ENDWHILE

ENDIF

 See Python/VB programs W6 Task 3 Theatre seats.py/vb

(a) Explain what line 2 of the program does:

 theatreFile = OPEN(fileName,"a")

Opens the file in Append mode ready for data to be added to the end of the existing data

(b) Will the program crash if the user enters an invalid date? Explain your answer.

 No – the exception handling routine will trap errors

(c) The user enters **audience2016.txt** as the name of the file in which to write the data.

 She then enters the following for dates and the number of seats sold on different dates:

 **Performance date Seats sold**

 111215 156

 121215 sdf

 200

 1231215 250

 xxx

 What happens when the user enters **sdf** for the number of seats sold?

 The program accepts the user entry, as this is a text file and the date is entered as a string.

(d) Trace through the program and say what the output will be in each of the cases below:

(i) when asked for the file name to read the data back from in Part 2 of the program, she enters

 audience2016

 **Program output:**

 File not found … program ending

 (ii) when asked for the file name to read the data back from, she enters

 audience2016.txt

 **Program output:**

 Date: 111215 Seats sold: 156

 Date: 121215 Seats sold: 200

 Date: 1231215 Seats sold: 250

 (e) What would be the simplest way to correct any invalid data in the test file?

Correct it in a text editor such as Notepad