Lab 7P: Files and Exceptions_ 7P_FE

Programming Exercises: GADDIS PYTHON 2e Text, Pages 292 – 294

Assigned Projects = 3 (65 Pts)

Using the IDLE editor (FILE/New Window)……

Name: _____Jeff Couch_____

For EACH Project you complete:

1. Create a FLOWCHART for each program and attach a Screenshot(s) of your FLOWCHART.

2. Take one Screenshot of both the EDIT window disiplaying your program and the SHELL window displaying your program results and paste as directed.

Project 1: File Display (20 Pts)

Input; numbers.txt

1. File Display

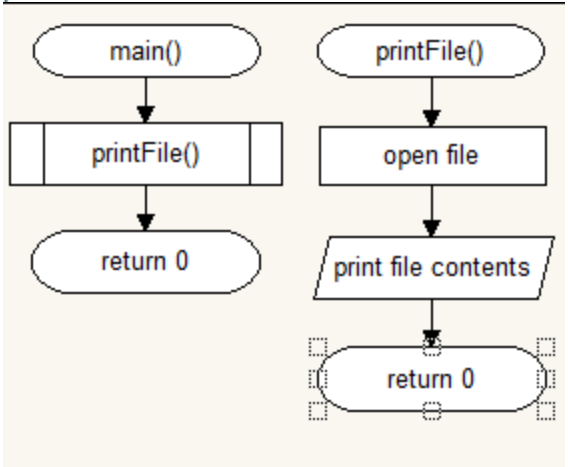
Assume that a file containing a series of integers is named `numbers.txt` and exists on the computer's disk. Write a program that displays all of the numbers in the file.

```
#1 File Display
def printFile(fileName):
    print(open(fileName, 'r').read(), end='')

def main():
    printFile('numbers.txt')

main()
```

```
>>>
110
85
25
85
1100
500
400
250
850
7500
999
2001
75
95
105
>>>
```



Project 6: Average of Numbers (20 Pts)

Input: numbers.txt

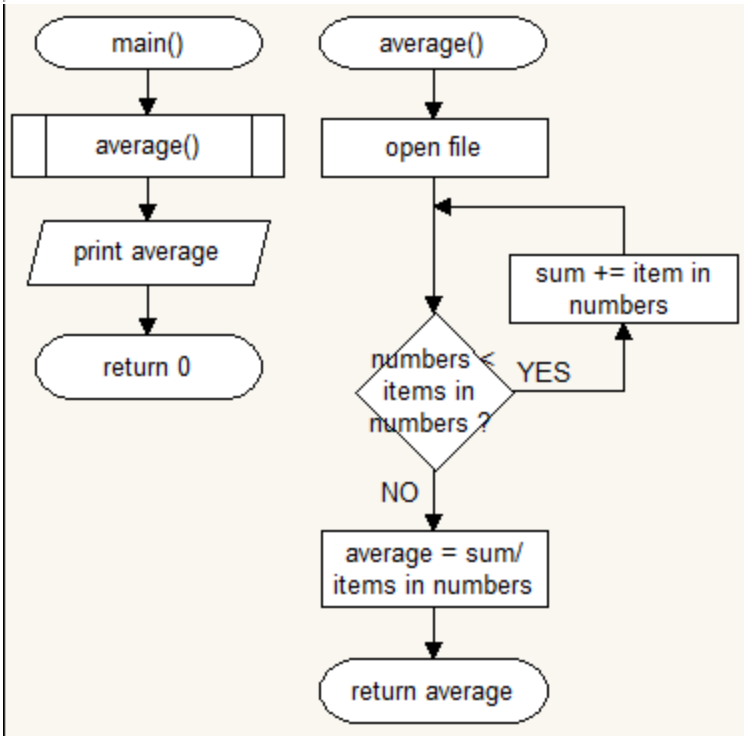
6. Average of Numbers

Assume that a file containing a series of integers is named `numbers.txt` and exists on the computer's disk. Write a program that calculates the average of all the numbers stored in the file.

```python
#6 Average of Numbers
def average(fileName):
    sumNums= 0
    with open(fileName) as numFile:
        numbers = numFile.read().split()
        for x in numbers:
            sumNums += int(x)
        average = float(sumNums)/len(numbers)
        return average

def main():
    print('The average is %.2f.' % average('numbers.txt'))

main()
```



Project 9: Exception Handling (25 Pts)

9. Exception Handing

Modify the program that you wrote for Exercise 6 so it handles the following exceptions:

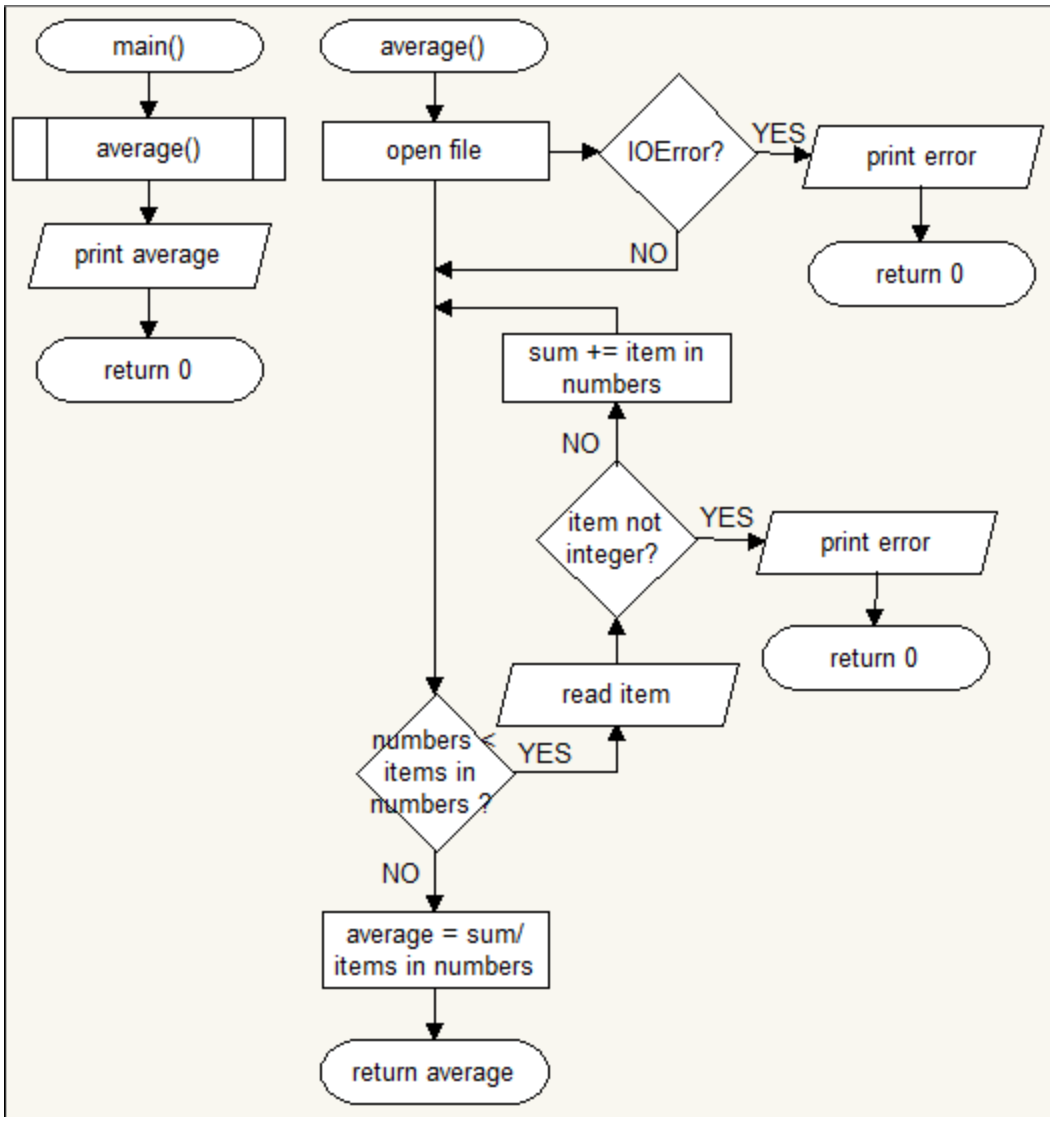
- It should handle any `IOError` exceptions that are raised when the file is opened and data is read from it.
- It should handle any `ValueError` exceptions that are raised when the items that are read from the file are converted to a number.

```python
#9 Exception Handling
def average(fileName):
    sumNums = 0
    try:
        with open(fileName) as numFile:
            numbers = numFile.read().split()
            for x in numbers:
                sumNums += int(x)
            average = float(sumNums)/len(numbers)
            return average
    except IOError:
        print('Could not read the file', fileName)
    except ValueError:
        print('Non-numeric data found in the file.')

def main():
    try:
        print('The average is %.2f.' % average('numbers.txt'))
    except:
        print('An error occurred.')
main()
```

```
110
85
25
85
1100
500
400
250
850
7500
999
2001
75
95
105
test
```

```
>>>
The average is 945.33.
>>> ================================ RESTART ================================
>>>
Non-numeric data found in the file.
An error occurred.
>>>
```


main()
average()
print average
return 0
average()
open file
IOError?
YES
print error
return 0
NO
sum += item in numbers
NO
item not integer?
YES
print error
return 0
read item
numbers < items in numbers ?
YES
NO
average = sum/ items in numbers
return average