

SAMPLE TEST AVG PROGRAM ORGANIZED AS FUNCTIONS:

File Edit Format Run Options Window Help

```
1 # Programmer: First & Last name
2 # Program Name: Creating the name array / list
3 # Date Written: March 25, 2020
4 # Purpose: Demonstrate how to define, create and display an array with test averages
5 # define the number of items that will be stored inside the list/array;
6 # Create a variety of functions that can be reused in this program or any programs
7 #=====
8 # Define the main program / module function
9 def main():
10
11     #Initialize Variables
12     average = 0.0
13     fileName = ''
14     outFile = ''
15     #=====
16     # Call function to create an output file to store results
17     fileName, outFile = createOutputFile()
18
19     #=====
20     print("How many test averages will be entered into this list? ")
21
22     # Call checkIntDataType function
23     numItems = checkIntDataType()
24
25     #=====
26     # count is the low/index
27     # Declare the array/list testAvg
28     testAvg = [0] * (numItems)
29
30     # initialize low count
31     count = 0
32
33     #=====
34     # Call function to create a list for Test Average
35     label = "Test Average # "
36     createList(numItems, label, testAvg)
37
38     #=====
39     # display heading
40     outFile.write("=" * 75 + "\n")
41     outFile.write(f'{"UNSORTED TEST AVERAGE LIST":^75s}' + "\n" )
42     outFile.write("=" * 75 + "\n")
43     outFile.write(" " + "\n")
44
45     # Reset array / list index back to 0
46     count = 0
47
48     # Call the function to write the list of Test averages
49     writeList(numItems, testAvg, outFile)
50
51     #=====
52     # sort the list using the predefined method "sort"
53     testAvg.sort()
54
55     #=====
56     # Call function to Calculate the test average
57     average = calcAvg(testAvg)
58
59     #=====
60
61     # Call function to write Statistics for test Average list
62     writeStats(average, testAvg, outFile)
63
64     #=====
65     # display heading
66     outFile.write("=" * 75 + "\n")
67     outFile.write(f'{"SORTED TEST AVERAGE LIST":^75s}' + "\n")
68     outFile.write("=" * 75 + "\n")
69     outFile.write(" " + "\n")
70
71     # Call the function to write the list of Test averages
72     writeList(numItems, testAvg, outFile)
73
74     #=====
75     # Call function to write Statistics for test Average list
76     writeStats(average, testAvg, outFile)
77
78     #=====
79
80     # Close the external file
81     outFile.close()
```

```

82
83 =====
84 # Define function to create the external file to store output
85 def createOutputFile():
86     # Create an external data file using functions/methods such as open, write and close
87     # to write results to an external output file
88     fName = input("Enter a file name to write the results to an output file\n")
89     oFile = open(fName + ".txt", "w")
90     return fName, oFile
91     # end createOutputFile function
92
93 =====
94 # Define function to check for integer input and for a positive whole integer (number)
95 def checkIntDataType():
96     while True:
97         try:
98             intDataType = int(input())
99         except ValueError:
100             print("You entered the wrong data type! - Re-enter a positive Whole number < no decimals>")
101             continue
102         else:
103             if intDataType <= 0:
104                 print("You entered a negative number! - Re-enter a positive Whole number < no decimals>")
105                 continue
106             else:
107                 break
108             # end if
109         # end try/except statement
110     # end while True loop
111     return intDataType
112     # end checkIntDataType function
113
114 =====
115 # Define function to check for real/float input and for a positive (number with decimal)
116 def checkFloatDataType():
117     while True:
118         try:
119             floatDataType = float(input())
120         except ValueError:
121             print("You entered the wrong data type! - Re-enter a positive numbers <decimals may be used>")
122             continue
123         else:
124             if floatDataType <= 0:
125                 print("You entered a negative number! - Re-enter a positive number <decimals may be used>")
126                 continue
127             else:
128                 break
129             # end if
130         # end try/except statement
131     # end while True loop
132     return floatDataType
133     # end checkFloatDataType function
134 =====

```

```

135 # Define function to write the temperature list
136 def writelist(size, tlist, oFile):
137     # Reset array / list index back to 0
138     count = 0
139
140     # While loop to display the items inside the list or array
141     while count < size:
142         oFile.write(format('{tList[count]:6.2f}'), "^75s") + "\n")
143
144         # update lcv/index
145         count = count + 1
146         # end while loop
147
148     oFile.write("=" * 75 + "\n")
149     # end writelist function
150
151
152 =====
153 # Define Function to calculate Average
154 def calcAvg(tList):
155     # Calculate the test average
156     result = sum(tList) / len(tList)
157     return result
158     # end calcAvg function
159
160 =====
161 # Define Function to write test average Statistics
162 def writeStats(avg, tList, oFile):
163
164     # Functions to find the minimum and maximum item(s) in the list using the predefined functions min and max
165     oFile.write("~" * 75 + "\n")
166     oFile.write(format('{min(tList):6.2f}{" %"}'), "^75s") + "\n")
167     oFile.write("~" * 75 + "\n")
168     oFile.write(format('{max(tList):6.2f}{" %"}'), "^75s") + "\n")
169     oFile.write("~" * 75 + "\n")
170     oFile.write(format('{avg:6.2f}{" %"}'), "^75s") + "\n")
171     oFile.write("~" * 75 + "\n")
172     # end writeStats function
173
174 =====
175 # Define function to create a list (one dimensional)
176 def createlist(size, tlabel, tlist):
177
178     # initialize lcv count
179     count = 0
180
181     # while loop to create a list
182     while count < size:
183         print(f'{tlabel} {(count + 1):2d} {" :"}')
184
185         # Call checkFloatDataType function
186         tlist[count] = checkFloatDataType()
187
188         # update lcv
189         count = count + 1
190         # end while loop
191     # end createlist function
192
193 =====
194 # Call the main function as a standalone program
195 if __name__ == '__main__':
196     main()
197
198 # END PROGRAM
199

```

SAMPLE TESTAVG PROGRAM SEPARATED AS THE SOURCE CODE PROGRAM AND A CUSTOMIZED PROGRAM WITH FUNCTIONS:

SOURCE CODE PROGRAM:

File Edit Format Run Options Window Help

```
1 # Programmer: First & Last name
2 # Program Name: Creating the name array / list
3 # Date Written: March 25, 2020
4 # Purpose: Demonstrate how to define, create and display an array with test averages
5 # define the number of items that will be stored inside the list/array;
6 # Create a variety of functions that can be reused in this program or any programs
7 #=====
9 # importing customised functions from the module lastname_firstname_myCustomFunctions
10 from lastname_firstname_myCustomFunctions import *
11
12 # Define the main program / module function
13 def main():
14
15     #Initialize Variables
16     average = 0.0
17     fileName = ''
18     outFile = ''
19     #=====
20     # Call function to create an output file to store results
21     fileName, outFile = createOutputFile()
22
23     #=====
24     print("How many test averages will be entered into this list? ")
25
26     # Call checkIntDataType function
27     numItems = checkIntDataType()
28
29     #=====
30     # count is the lcv/index
31     # Declare the array/list testAvg
32     testAvg = [0] * (numItems)
33
34     #=====
35     # Call function to create a list for Test Average
36     label = "Test Average # "
37     createList(numItems, label, testAvg)
38
39     #=====
40     # display heading
41     outFile.write("=" * 75 + "\n")
42     outFile.write(f'{"UNSORTED TEST AVERAGE LIST":^75s}' + "\n")
43     outFile.write("=" * 75 + "\n")
44     outFile.write(" " + "\n")
45
46     # Call the function to write the list of Test averages
47     writeList(numItems, testAvg, outFile)
48
49     #=====
50     # sort the list using the predefined method "sort"
51     testAvg.sort()
52
53     #=====
54     # Call function to Calculate the test average
55     average = calcAvg(testAvg)
56
57     #=====
58
59     # Call function to write Statistics for test Average list
60     writeStats(average, testAvg, outFile)
61
62     #=====
63     # display heading
64     outFile.write("=" * 75 + "\n")
65     outFile.write(f'{"SORTED TEST AVERAGE LIST":^75s}' + "\n")
66     outFile.write("=" * 75 + "\n")
67     outFile.write(" " + "\n")
68
69     # Call the function to write the list of Test averages
70     writeList(numItems, testAvg, outFile)
71
72     #=====
73     # Call function to write Statistics for test Average list
74     writeStats(average, testAvg, outFile)
75
76     #=====
77
78     # Close the external file
79     outFile.close()
80     # end main function
81     #=====
82
83 #=====
84 # Call the main function as a standalone program
85 if __name__ == '__main__':
86     main()
87
88 # END PROGRAM
```

lastname_firstname_MyCustom_Functions program containing the functions.

File Edit Format Run Options Window Help

```
1 # PROGRAMMER: first and last name
2 # PROGRAM NAME: lastname_firstname_myCustomFunctions
3 # DATE WRITTEN: June 17, 2021
4 #=====
5
6 # Define function to create the external file to store output
7 def createOutputFile():
8     # Create an external data file using functions/methods such as open, write and close
9     # to write results to an external output file
10    fName = input("Enter a file name to write the results to an output file\n")
11    oFile = open(fName + ".txt", "w")
12    return fName, oFile
13    # end createOutputFile function
14
15 #=====
16 # Define function to check for integer input and for a positive whole integer (number)
17 def checkIntDataType():
18     while True:
19         try:
20             intDataType = int(input())
21         except ValueError:
22             print("You entered the wrong data type! - Re-enter a positive Whole number < no decimals>")
23             continue
24         else:
25             if intDataType <= 0:
26                 print("You entered a negative number! - Re-enter a positive Whole number < no decimals>")
27                 continue
28             else:
29                 break
30             # end if
31         # end try/except statement
32     # end while True loop
33     return intDataType
34     # end checkIntDataType function
35
36 #=====
37 # Define function to check for real/float input and for a positive (number with decimal)
38 def checkFloatDataType():
39     while True:
40         try:
41             floatDataType = float(input())
42         except ValueError:
43             print("You entered the wrong data type! - Re-enter a positive numbers <decimals may be used>")
44             continue
45         else:
46             if floatDataType <= 0:
47                 print("You entered a negative number! - Re-enter a positive number <decimals may be used>")
48                 continue
49             else:
50                 break
51             # end if
52         # end try/except statement
53     # end while True loop
54     return floatDataType
55     # end checkFloatDataType function
56 #=====
57 # Define function to write the temperature list
58 def writeList(size, tList, oFile):
59     # Reset array / list index back to 0
60     count = 0
61
62     # While loop to display the items inside the list or array
63     while count < size:
64         oFile.write(format('{{tList[count]:6.2f}}', "^75s") + "\n")
65
66         # update low/index
67         count = count + 1
68     # end while loop
69
70     oFile.write("=" * 75 + "\n")
71     # end writeList function
72
73
74 #=====
```

```

75 # Define Function to calculate Average
76 def calcAvg(tlist):
77     # Calculate the test average
78     result = sum(tlist) / len(tlist)
79     return result
80     # end calcAvg function
81
82 =====
83 # Define Function to write test average Statistics
84 def writeStats(avg, tlist, ofile):
85
86     # Functions to find the minimum and maximum item(s) in the list using the predefined functions min and max
87     ofile.write("~" * 75 + "\n")
88     ofile.write(format((f'{"The minimum test average in the list is "}{min(tlist):6.2f}{" %"}'), "^75s") + "\n")
89     ofile.write("~" * 75 + "\n")
90     ofile.write(format((f'{"The maximum test average in the list is "}{max(tlist):6.2f}{" %"}'), "^75s") + "\n")
91     ofile.write("~" * 75 + "\n")
92     ofile.write(format((f'{"The Average of the test scores in the list is "}{avg:6.2f}{" %"}'), "^75s") + "\n")
93     ofile.write("~" * 75 + "\n")
94     # end writeStats function
95
96 =====
97 # Define function to create a list (one dimensional)
98 def createlist(size, tlabel, tlist):
99
100     # initialise low count
101     count = 0
102
103     # while loop to create a list
104     while count < size:
105         print(f'{tlabel} {(count + 1):2d} {" ": 1}')
106
107         # Call checkFloatDataType function
108         tlist[count] = checkFloatDataType()
109
110         # update low
111         count = count + 1
112     # end while loop
113     # end createlist function
114
115 =====
116 # END PROGRAM [FUNCTION DEFINITIONS]
117

```

MAKE SURE THAT THE SOURCE CODE PROGRAM, TESTAVG is stored in the same folder were the lastname_firstname_myCustomFunctions.py are stored. BOTH MUST BE INSIDE THE SAME FOLDER.