Matthew Ojeda
IS 340
M & W 11am-12:15pm

HW3

Question 1

```
from statistics import *
#start by asking how many students will you be inputting grades for
noStdts = int(input("Enter number of students: "))

#Ask for input of hw1
hw1Grades=[]
for i in range (noStdts):
    hw1Input = float(input("Enter Grade for student no." +str(i)+ " for Hw 1 out of 20 (Enter -1 to
stop): "))
    hw1Grades.append(hw1Input)
    if hw1Input<0:
        hw1Grades.pop(i)
        break
    elif hw1Input>20 or hw1Input==str:
        print("Invalid entry")

#ask for input of hw2
hw2Grades=[]
for i in range (noStdts):
    hw2Input = float(input("Enter Grade for student no." +str(i)+ " for Hw 2 out of 20 (Enter -1 to
stop): "))
    hw2Grades.append(hw2Input)
    if hw2Input<0:
        hw2Grades.pop(i)
        break
    elif hw2Input>20 or hw2Input==str:
        print("Invalid entry")

#Ask for input of hw3
hw3Grades=[]
for i in range (noStdts):
    hw3Input = float(input("Enter Grade for student no." +str(i)+ " for Hw 3 out of 20 (Enter -1 to
stop): "))
    hw3Grades.append(hw3Input)
    if hw3Input<0:
        hw3Grades.pop(i)
        break
    elif hw3Input>20 or hw3Input==str:
```

```
        print("Invalid entry")

#Ask for input of Midterm
midtermGrades=[]
for i in range (0, noStdts):
    midInput = float(input("Enter Grade for student no." +str(i)+ " for Midterm exam out of 20
(Enter -1 to stop): "))
    midtermGrades.append(midInput)
    if midInput<0:
        midtermGrades.pop(i)
        break
    elif midInput>20 or midInput==str:
        print("Invalid entry")

#Ask for input of final exam
finalGrades=[]
for i in range (0, noStdts):
    finalInput = float(input("Enter Grade for student no." +str(i)+ " for final exam out of 20 (Enter -1
to stop): "))
    finalGrades.append(finalInput)
    if finalInput<0:
        finalGrades.pop(i)
        break
    elif finalInput>20 or finalInput==str:
        print("Invalid entry")
#Create some space
print()
#Create a list within a list
workbook = [hw1Grades, hw2Grades, hw3Grades, midtermGrades, finalGrades]
#You can use the statistics library for these calculations
#Find mode with function
def mode(listTitle):
    listTitle.sort()
    counter=0
    modes=0
    for i in listTitle:
        if listTitle.count(i)>counter:
            counter=listTitle.count(i)
            modes=i
    return modes
#find average for hw1
hw1Avg = mean(hw1Grades)
#find the min of hw1
```

```
minHw1 = min(hw1Grades)
#max for hw1
maxHw1 = max(hw1Grades)
#find median for hw1, have to sort first
hw1Grades.sort()
medHw1 = median(hw1Grades)
#find mode for hw1
modeHw1 = mode(hw1Grades)
#find std dev from hw1
hw1Stdev = stdev(hw1Grades)

#find average for hw2
hw2Avg = mean(hw2Grades)
#find the min of hw2
minHw2 = min(hw2Grades)
#max for hw2
maxHw2 = max(hw2Grades)
#find median for hw2, have to sort first
hw2Grades.sort()
medHw2 = median(hw2Grades)
#find mode for hw2
modeHw2 = mode(hw2Grades)
#find std dev from hw2
hw2Stdev = stdev(hw2Grades)

#find average for hw3
hw3Avg = mean(hw3Grades)
#find the min of hw3
minHw3 = min(hw3Grades)
#max for hw3
maxHw3 = max(hw3Grades)
#find median for hw3, have to sort first
hw3Grades.sort()
medHw3 = median(hw3Grades)
#find mode for hw3
modeHw3 = mode(hw3Grades)
#find std dev from hw3
hw3Stdev = stdev(hw3Grades)

#find average for midterm
midAvg = mean(midtermGrades)
#find the min of midterm
minMid = min(midtermGrades)
```

```
#max for midterm
maxMid = max(midtermGrades)
#find median for midterm, have to sort first
midtermGrades.sort()
medMid = median(midtermGrades)
#find mode for midterm
modeMid = mode(midtermGrades)
#find std dev from midterm
midStdev = stdev(midtermGrades)

#find average for final
finAvg = mean(finalGrades)
#find the min of midterm
minFin = min(finalGrades)
#max for midterm
maxFin = max(finalGrades)
#find median for midterm, have to sort first
finalGrades.sort()
medFin = median(finalGrades)
#find mode for midterm
modeFin = mode(finalGrades)
#find std dev from midterm
finStdev = stdev(finalGrades)

#Organize the mean,median,mode,max,min, and std dev neatly to be displayed
print("%27s%10s%10s%12s%10s" %("Hw1", "Hw2", "Hw3", "Midterm", "Final"))
print("Average: ", end="")
print("%20.2f%10.2f%10.2f%10.2f%10.2f" %(hw1Avg, hw2Avg, hw3Avg, midAvg, finAvg))
print("Minimum: ", end="")
print("%20.2f%10.2f%10.2f%10.2f%10.2f" %(minHw1, minHw2, minHw3, minMid, minFin))
print("Maximum: ", end="")
print("%20.2f%10.2f%10.2f%10.2f%10.2f" %(maxHw1, maxHw2, maxHw3, maxMid, maxFin))
print("Median: ", end="")
print("%21.2f%10.2f%10.2f%10.2f%10.2f" %(medHw1, medHw2, medHw3, medMid, medFin))
print("Mode: ", end="")
print("%23.2f%10.2f%10.2f%10.2f%10.2f" %(modeHw1, modeHw2, modeHw3, modeMid,
modeFin))
print("Std Dev: ", end="")
print("%20.2f%10.2f%10.2f%10.2f%10.2f" %(hw1Stdev, hw2Stdev, hw3Stdev, midStdev,
finStdev))
```

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\matto_000\Documents\IS 340\OjedaHW3Q1.py =========
Enter number of students: 3
Enter Grade for student no.0 for Hw 1 out of 20 (Enter -1 to stop): 15
Enter Grade for student no.1 for Hw 1 out of 20 (Enter -1 to stop): 14
Enter Grade for student no.2 for Hw 1 out of 20 (Enter -1 to stop): 10
Enter Grade for student no.0 for Hw 2 out of 20 (Enter -1 to stop): 16
Enter Grade for student no.1 for Hw 2 out of 20 (Enter -1 to stop): 15
Enter Grade for student no.2 for Hw 2 out of 20 (Enter -1 to stop): 14
Enter Grade for student no.0 for Hw 3 out of 20 (Enter -1 to stop): 20
Enter Grade for student no.1 for Hw 3 out of 20 (Enter -1 to stop): 11
Enter Grade for student no.2 for Hw 3 out of 20 (Enter -1 to stop): 20
Enter Grade for student no.0 for Midterm exam out of 20 (Enter -1 to stop): 15
Enter Grade for student no.1 for Midterm exam out of 20 (Enter -1 to stop): 18
Enter Grade for student no.2 for Midterm exam out of 20 (Enter -1 to stop): 19
Enter Grade for student no.0 for final exam out of 20 (Enter -1 to stop): 11
Enter Grade for student no.1 for final exam out of 20 (Enter -1 to stop): 16
Enter Grade for student no.2 for final exam out of 20 (Enter -1 to stop): 20

                      Hw1       Hw2       Hw3     Midterm     Final
Average:            13.00     15.00     17.00       17.33     15.67
Minimum:            10.00     14.00     11.00       15.00     11.00
Maximum:            15.00     16.00     20.00       19.00     20.00
Median:             14.00     15.00     20.00       18.00     16.00
Mode:               10.00     14.00     20.00       15.00     11.00
Std Dev:             2.65      1.00      5.20        2.08      4.51
>>>
```

Ln: 29 Col: 4

Question 2

```
#set a constant for how many rows and columsn there will be(country=rows, medals=columns)
COUNTRIES = 5
MEDALS = 3
#create the list with the country names
countries = ["USA",
             "China",
             "Russia",
             "Spain",
```

```
            "UK"]
#create a list of the medal counts
counts = [
   [5, 6 ,3],
   [4, 7, 3],
   [3, 2, 1],
   [2, 1, 2],
   [1, 1, 1]
   ]
#make a seperate list for total rows
totalRow=[5,6,3,4,7,3,3,2,1,2,1,2,1,1,1]
#print the table header
print("%10s%8s%8s%8s%8s" %("Country", "Gold", "Silver", "Bronze", "Total"))
#now to print the countries, counts, and row totals
for i in range (COUNTRIES):
   print("%10s" %countries[i], end="")

   #print each row element and get the totals
   total = 0
   for j in range (MEDALS):
      print("%7d" %counts[i][j], end="")
      total= total + counts[i][j]
   #Show the row totals
   print("%10s" %total)
#Show column totals
print("%10s" %("Total"), end="")
for j in range (MEDALS):
   sumColumn = 0
   for i in range (COUNTRIES):
      sumColumn = sumColumn + counts[i][j]
   print("%7d" %sumColumn, end="")
addedQ = 0
addedQ = sum(totalRow)
print("%10d" %addedQ)

#print average
print("%10s" %("Average"), end="")
#create a function for finding average
def medalAvg(totalN, number):
   avg= float(totalN / number)
   return avg
addedAvg = float(medalAvg(addedQ, COUNTRIES))
for m in range (MEDALS):
```

```
    sumColumn=0
    for p in range (COUNTRIES):
        sumColumn = sumColumn + counts[p][m]
        average= float(medalAvg(sumColumn, COUNTRIES))
    print("%8.2f" %average, end="")
print("%7.2f" %addedAvg)
```

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\matto_000\Documents\IS 340\OjedaHW3Q2.py =========
   Country    Gold  Silver  Bronze   Total
       USA       5       6       3      14
     China       4       7       3      14
    Russia       3       2       1       6
     Spain       2       1       2       5
        UK       1       1       1       3
     Total      15      17      10      42
   Average    3.00    3.40    2.00    8.40
>>>
```