Introduction to Programming with Python

Weekly Programming Assignment – Week 9

All solution files must be submitted at CodeGrade enabled Link for grading.
All solutions must be uploaded on or before 17th November 2021 at 11:59 PM

Exercise 1

Write a program that accepts only positive float numbers as input from the user until user enter **0** to terminate the program. Then the accepted values must be displayed in ascending order. However, if the user input is negative number or other data type then **ValueError Exception** (from Python's library) must be thrown with message "*Give only positive float numbers*" and continue asking input from the user again. The sample output is here:

```
>>> %Run Ex1_Week9.py
 Enter a float value:0
>>> %Run Ex1_Week9.py
 Enter a float value:23
 Give only positive float numbers
 Enter a float value:-15.6
 Give only positive float numbers
 Enter a float value:89.0
 Enter a float value:34.7
 Enter a float value:asd
 Give only positive float numbers
 Enter a float value:4.90
 Enter a float value:0
 4.9
 34.7
 89.0
```

Exercise 2

Write a procedure **checkFileType()** that accepts any filename as argument and prints its contents. However, appropriate error message *caught by Python exception class* must be displayed if file does not exist [Should use Python's Exception only]. The message "**The program ends**" must be displayed despite coding errors. Your code must use **try-catch-else-finally** to execute this task. The sample run is here.

```
13 #main program
14 checkFileType("person.txt")
15 checkFileType("person1.txt")
Shell
Python 3.7.9 (bundled)
>>> %Run Ex2_Week9.py
 Ali:56
 Wali:28a
 Ram:16
 Vijay:56
 Abdur:26
 Chen:18.02
 Wang:
 Joy:21
 Fatimah:22
 Anita:abc
 Fritz:"
 The program ends
 [Errno 2] No such file or directory: 'person1.txt'
 The program ends
```

Exercise 3

The file **person.txt** contain names and their ages. Write a procedure **checkAgeEntry()** that accepts any filename as input and checks whether the ages of all persons are entered correctly. That is, if the age is missing/in float form/or in other data type then your code should throw appropriate message with name of the person and continue checking next one [use **ValueError]**. The main program is given below to test your procedure.

```
13 #main program
14 checkAgeEntry("person.txt")
```

```
Shell
Python 3.7.9 (bundled)
>>> %Run Ex3_Week9.py
 age entry is correct Ali 56
 age entry is not correct Wali 28a
 age entry is correct Ram 16
 age entry is correct Vijay 56
 age entry is correct Abdur 26
 age entry is not correct Chen 18.02
 age entry is not correct Wang
 age entry is correct Joy 21
 age entry is correct Fatimah 22
 age entry is not correct Anita abc
 age entry is not correct Fritz "
```

Exercise 4

Write a program that accepts customer name, age, and salary. Then write that information into a file "**customer.txt**" until user wants to. However, custom exceptions must be used to display appropriate messages, if

i. any of the user input is not correct (data type) – use python's exception
ii. salary input is not within 1500.0 and 2500.0 euros -define custom
iii. at least 25 years old – define custom exception

if any of the input is wrong then the user must prompt to enter all information again (may give different input as well). Finally, the contents of the file must be displayed. The sample run is here.

```
Python 3.7.9 (bundled)
>>> %Run Ex4_Week9.py
 Your name:Ashok
 Your salary:2300
 Your age:17
 age must be 25 or more:17
 Your name:Ashok
 Your salary:2300
 Your age:28
 do you want to continue y or n:y
 Your name:Kumar
 Your salary:1450
 salary must be between 1500 and 2500:1450.0
 Your name:Kumar
 Your salary:1780
 Your age:29
 do you want to continue y or n:y
 Your name:Wali
 Your salary:dqwe
 could not convert string to float: 'dqwe'
 Your name:Wali
 Your salary:2400
 Your age:asdqw
 invalid literal for int() with base 10: 'asdqw'
 Your name:Wali
 Your salary:2400
 Your age:56
 do you want to continue y or n:n
 Ashok,2300.0,28
 Kumar,1780.0,29
 Wali,2400.0,56
```

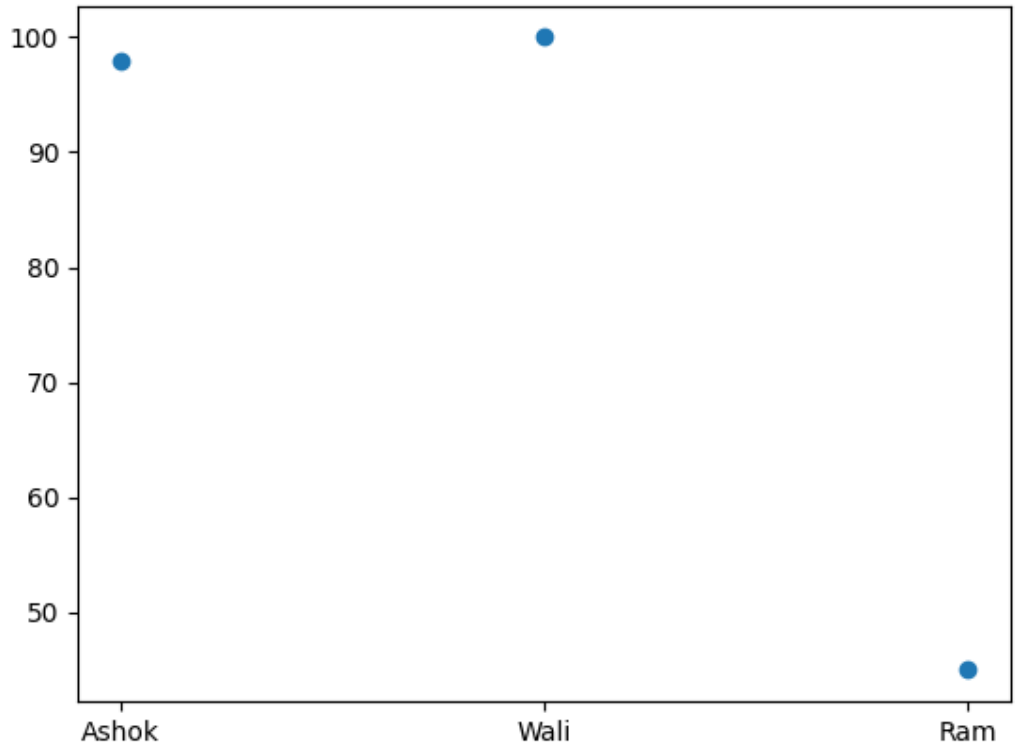
Define custom exceptions namely **salaryNotinRangeError** for salary check, **ageNotinRangeError** for age check, and Python's V**alueError** for input check of all inputs to proceed further. You are free to define subprograms to get the aforenoted results. [Hint: lecture slides 6-9]

Exercise 5 (self-study and show during tutorial /lecture times)

The program given below represents how to make simple dot plot of given set of x and y values.

```
1 import matplotlib.pyplot as plt # library to plot graph
2 import numpy as np # it should be installed first
3 xpoints = np.array(["Ashok","Wali","Ram"]) # x points
4 ypoints = np.array([98,100,45]) # y points
5
6 plt.plot(xpoints, ypoints,'o') # o means filled circle
7 plt.show() # show the graph
8
```

Graph output



Refer the given program above to write program for the following.

Write a program that accept any 5 persons names (key) and scores(values) as input and update those in the dictionary. Then display the items of dictionary as graph like given above (*but you are free to use any graph- like bar, line graph→ be more creative or think out of the box). Keep in mind inputs must be checked by valid exceptions before getting into the dictionary. If the given input is wrong throw **an error message**, then user must be prompt to enter the input again until dictionary gets all 5 items. Submit your final code file as **.txt** and may be asked to run the code during tutorial sessions. ***Surprise gift is hidden**. All the best.

Exercise / task Number	Codegrade link_Moodle for file solution files upload	Points / Marks
1	Exercise1_Week 9	15
2	Exercise2_ Week 9 + person.txt	10
3	Exercise3_Week 9 + person.txt	15
4	Exercise4_Week 9	20
5	Exercise5_Week 9 – Moodle submission as txt file	10