

Assignment 1

Name: Mythresh Maddina

Uid No: 700741162

Video link:

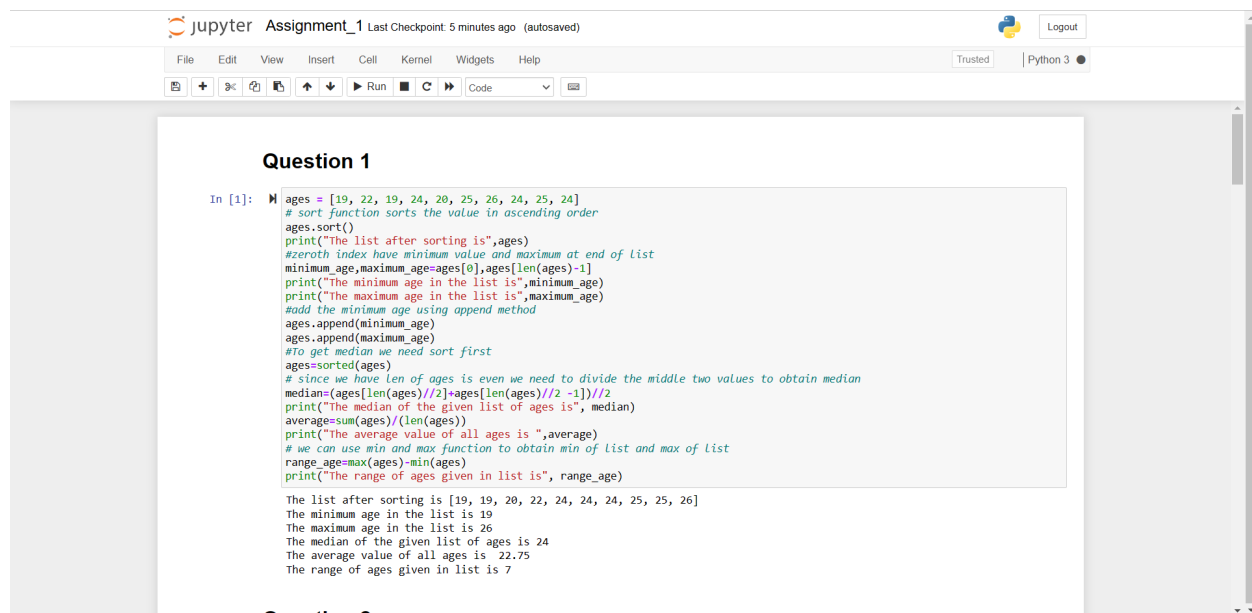
https://drive.google.com/file/d/1BuBz2YrbX37-F_I_Da-DKmf3rK_29G7Q/view?usp=share_link

Github Link: https://github.com/MythreshM/CS5710_Assignment1

Document Link:

<https://docs.google.com/document/d/1Kuq-Su6bbsePuQEmh5sJ7ymqP4uwFyHDGTJEtgJaTms/edit?usp=sharing>

Question 1:



```
jupyter Assignment_1 Last Checkpoint: 5 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [1]: ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 26]
# sort function sorts the value in ascending order
ages.sort()
print("The list after sorting is", ages)
# zeroth index have minimum value and maximum at end of list
minimum_age, maximum_age = ages[0], ages[len(ages)-1]
print("The minimum age in the list is", minimum_age)
print("The maximum age in the list is", maximum_age)
# add the minimum age using append method
ages.append(minimum_age)
ages.append(maximum_age)
# to get median we need sort first
ages = sorted(ages)
# since we have len of ages is even we need to divide the middle two values to obtain median
median = (ages[len(ages)//2] + ages[len(ages)//2 - 1]) / 2
print("The median of the given list of ages is", median)
average = sum(ages) / (len(ages))
print("The average value of all ages is ", average)
# we can use min and max function to obtain min of list and max of list
range_age = max(ages) - min(ages)
print("The range of ages given in list is", range_age)

The list after sorting is [19, 19, 20, 22, 24, 24, 25, 25, 26]
The minimum age in the list is 19
The maximum age in the list is 26
The median of the given list of ages is 24
The average value of all ages is 22.75
The range of ages given in list is 7
```

Question 2:

jupyter Assignment_1 Last Checkpoint 6 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Question 2

```
In [2]: #creating a dog dictionary
dog={}
dog["name"]="Snoopy"
dog["color"]="Black"
dog["breed"]="German Shepherd"
dog["legs"]=4
dog["age"]=5
student={"first_name":"Rajesh","last_name":"Nalla", "gender":"male","age":22,"marital_status":"Single","skills":["communication skills", "presentation skills"]}
# len method is used to find length of an dictionary
print("The length of student dictionary is",len(student))
print("The value of skills key is",student["skills"])
print("The datatype of skills value is list:", type(student["skills"]))
student["skills"].append("Essay Writing")
student["skills"].append("MS")
print("The student dictionary contains:",student)
#printing keys and value of dog dictionary as list
print("The keys of dog dictionary is",list(dog.keys()))
print("The values of dog dictionary is",list(dog.values()))
```

The length of student dictionary is 9
The value of skills key is ['communication skills', 'presentation skills']
The datatype of skills value is list: True
The student dictionary contains: {'first_name': 'Rajesh', 'last_name': 'Nalla', 'gender': 'male', 'age': 22, 'marital_status': 'Single', 'skills': ['communication skills', 'presentation skills', 'Essay Writing', 'MS'], 'country': 'India', 'city': 'Chennai', 'address': 'RSM Nagar, Gummidipoondi, Taluk, Kavarajpettai, Tamil Nadu 601206, India'}
The keys of dog dictionary is ['name', 'color', 'breed', 'legs', 'age']
The values of dog dictionary is ['Snoopy', 'Black', 'German Shepherd', 4, 5]

Question 3:

jupyter Assignment_1 Last Checkpoint 7 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Question 3

```
In [3]: #brothers=tuple()
brothers=brothers+("Arun","Sam")
print("The names of brothers are",brothers)
sisters=("Meghana",)
print("The name of sister is",sisters)
#add brothers and sisters to siblings using + operator
siblings=brothers+sisters
print("The siblings are:",siblings)
print("The number of siblings I have are:",len(siblings))
#Since one of the brother changed his name from Sam to Harsha need to change brothers tuple and siblings tuple
modified=[]
modified_brother=[]
for x in siblings:
    if x=="Sam":
        modified.append("Harsha")
    else:
        modified.append(x)
    if modified[len(modified)-1] not in sisters:
        modified_brother.append(modified[len(modified)-1])
#assign modified values to siblings
siblings=tuple(modified)
brothers=modified_brother
print("The names of the siblings after changing name is ",siblings)
#Assume name of father and mother is Ram and Sita respectively
father_name,mother_name=["Ram"],["Sita"]
family_members=tuple(father_name)+tuple(mother_name)+siblings
print("The members of the family are ",family_members)
```

The names of brothers are ('Arun', 'Sam')
The name of sister is ('Meghana',)
The siblings are: ('Arun', 'Sam', 'Meghana')
The number of siblings I have are: 3
The names of the siblings after changing name is ('Arun', 'Harsha', 'Meghana')
The members of the family are ('Ram', 'Sita', 'Arun', 'Harsha', 'Meghana')

Question 4:

Code:

```
jupyter Assignment_1 Last Checkpoint: 10 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Question 4

In [4]: it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
A = {19, 22, 24, 26, 25, 26}
B = {19, 22, 28, 25, 26, 24, 28, 27}
age = [22, 19, 24, 25, 26, 24, 25, 24]
#length of it_companies set
print("The length of it companies set is :",len(it_companies))
#Add twitter to it_companies
it_companies.add('Twitter')
print("The it companies set after adding Twitter to it is:",it_companies)
#insert multiple elements to set once
it_companies.update(['Kaar Technologies','Wipro','Infosys'])
print("The set of it companies are:",it_companies)
#Removing Kaar Technologies from set of it companies
it_companies.remove('Kaar Technologies')
print("The set after removing Kaar Technologies is:",it_companies)
#Difference between remove and discard
print("\nDifference between remove and discard
Remove raises keyError if the element not present in set else removes the element.

whereas

Discard removes the element but does not raise error even if we are trying to remove the element that is absent.\n
...')
#Join A and B
AB=A.union(B)
print("The value of set joining A and B is :",AB)
#Intersection of A and B
print("The Intersection of A and B is",A.intersection(B))
#Is A subset of B
print("Is A subset of B:",A.issubset(B))
#Disjoint sets
print("Is A and B are disjoint sets:",A.isdisjoint(B))
#Join A and B and then B and A
print("Joining A and B",A.union(B))
print("Joining B and A",B.union(A))
#Symmetric difference of A and B
print("The symmetric difference of A and B is:",A.symmetric_difference(B))
#Delete set A and B completely using clear method
A.clear()
B.clear()
print("The set A after clear method are {} and length is {}".format(A, len(A)))
print("The setB after clear method are {} and length is {}".format(B, len(B)))
#Convert ages list to set
set_ages=set(age)
print("The length of ages in list is {} and the length of ages set is {}".format(len(age),len(set_ages)))
```

Output:

```
jupyter Assignment_1 Last Checkpoint: 11 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Intersection of A and B
The Intersection of A and B is {24, 25, 26}
Is A subset of B
print("Is A subset of B:",A.issubset(B))
#Disjoint sets
print("Is A and B are disjoint sets:",A.isdisjoint(B))
#Join A and B and then B and A
print("Joining A and B",A.union(B))
print("Joining B and A",B.union(A))
#Symmetric difference of A and B
print("The symmetric difference of A and B is:",A.symmetric_difference(B))
#Delete set A and B completely using clear method
A.clear()
B.clear()
print("The set A after clear method are {} and length is {}".format(A, len(A)))
print("The setB after clear method are {} and length is {}".format(B, len(B)))
#Convert ages list to set
set_ages=set(age)
print("The length of ages in list is {} and the length of ages set is {}".format(len(age),len(set_ages)))

The length of it companies set is : 7
The it_companies set after adding Twitter to it is: {'Twitter', 'Facebook', 'Microsoft', 'IBM', 'Amazon', 'Apple', 'Oracle', 'Google'}
The set of it companies are: {'Wipro', 'Twitter', 'Facebook', 'Kaar Technologies', 'Microsoft', 'IBM', 'Amazon', 'Apple', 'Infosys', 'Oracle', 'Google'}
The set after removing Kaar Technologies is: {'Wipro', 'Twitter', 'Facebook', 'Microsoft', 'IBM', 'Amazon', 'Apple', 'Infosys', 'Oracle', 'Google'}

Difference between remove and discard
Remove raises keyError if the element not present in set else removes the element.

whereas

Discard removes the element but does not raise error even if we are trying to remove the element that is absent.

The value of set joining A and B is : {19, 20, 22, 24, 25, 26, 27, 28}
The Intersection of A and B is {19, 20, 22, 24, 25, 26}
Is A subset of B: True
Is A and B are disjoint sets: False
Joining A and B {19, 20, 22, 24, 25, 26, 27, 28}
Joining B and A {19, 20, 22, 24, 25, 26, 27, 28}
The symmetric difference of A and B is: {27, 28}
The set A after clear method are set() and length is 0
The setB after clear method are set() and length is 0
The length of ages in list is 8 and the length of ages set is 5.
```


Question 5:

Question 5

```
In [5]: # radius=30
        PI=3.14
        #compute area of circle
        area_of_circle=PI* radius*radius
        print("The area of circle is {} square meters.".format(area_of_circle))
        #compute circumference of circle
        circum_of_circle=2*PI*radius
        print("The circumference of circle is {} meters.".format(circum_of_circle))
        # radius as user input
        radii=float(input("Enter the radius: "))
        #compute area
        print("The area of circle for radii {} is {} square meters.".format(radii,PI*radii*radii))

The area of circle is 2826.0 square meters.
The circumference of circle is 188.4 meters.
Enter the radius: 20
The area of circle for radii 20.0 is 1256.0 square meters.
```

Question 6:

jupyter Assignment_1 Last Checkpoint: 14 minutes ago (unsaved changes)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

The area of circle for radii 20.0 is 1256.0 square meters.

Question 6

```
In [6]: # sentence="I am a teacher and I love to inspire and teach people"
        # use split and set
        words_set=set(sentence.split())
        # Length of unique words
        print("The number of unique words in the given sentence is :",len(words_set))

The number of unique words in the given sentence is : 10
```

Question 7:

Question 7

```
In [7]: #Here we use \t escape character to provide space tab between string
        print("Name", "Age", "Country", "City", sep='\t')
        print("Asabeneh", 250, "Finland", "Helsinki", sep='\t')
```

Name	Age	Country	City
Asabeneh	250	Finland	Helsinki

Question 8:

Question 8

```
In [8]: # radius = 10
        area = 3.14 * radius ** 2
        print("The area of a circle with radius {} is {} meters square.".format(radius,int(area)))

The area of a circle with radius 10 is 314 meters square.
```

Question 9:

Question 9

```
In [9]: N=int(input("Enter no of students: "))
weight_in_lbs=[int(input()) for i in range(N)]
print("L1 :",weight_in_lbs)
weight_in_kgs=[weight_in_lbs[i]*0.453592 for i in range(N)]
print("Output:",weight_in_kgs)

Enter no of students: 4
150
155
145
148
L1 : [150, 155, 145, 148]
Output: [68.0388, 70.30676, 65.77083999999999, 67.131616]
```

Question 10:

- Given there are 2 classes and 8 datasets with feature value f.
- The class with “O” is taken as zero and the class with “X” is taken as 1.

DataSet:

S.No	Feature Value	Class	ClassValue
1	1	O	0
2	2	O	0
3	3	X	1
4	6	X	1
5	6	X	1
6	7	O	0
7	10	O	0
8	11	O	0

1.

Put the first 4 data points as a training set.

Training Set:

S.No	Feature Value	Class	Class Value
1	1	O	0
2	2	O	0
3	3	X	1
4	6	X	1

Testing set:

S. No	Feature Value
1	6
2	7
3	10
4	11

- Here we use KNN classifier with value K value K=3

For Feature Value f=6:

- F=2, F=3 and F=6 are the nearest three neighbors.
- Now calculate distance b/w testing value and its neighbors.

Feature Value(F)	Distance
2	$(2-6)^2=4^2=16$
3	$(3-6)^2=3^2=9$

6	$(6-6)^2=0$
---	-------------

The feature value 6 is near to the testing data point. So, it is predicted to be class value 1.

For Feature Value $f=7$:

- $F=2$, $F=3$ and $F=6$ are the nearest three neighbors.
- Now calculate distance b/w testing value and its neighbors.

Feature Value(F)	Distance
2	$(2-7)^2=5^2=25$
3	$(3-7)^2=4^2=16$
6	$(6-7)^2=1$

The feature value 6 is near to the testing data point. So, it is predicted to be class value 1.

For Feature Value $f=10$:

- $F=2$, $F=3$ and $F=6$ are the nearest three neighbors.
- Now calculate distance b/w testing value and its neighbors.

Feature Value(F)	Distance
2	$(2-10)^2=8^2=64$
3	$(3-10)^2=7^2=49$
6	$(6-10)^2=4^2=16$

The feature value 6 is near to the testing data point. So, it is predicted to be class value 1.

For Feature Value $f=11$:

- $F=2$, $F=3$ and $F=6$ are the nearest three neighbors.
- Now calculate distance b/w testing value and its neighbors.

Feature Value(F)	Distance
2	$(2-11)^2=9^2=81$
3	$(3-11)^2=8^2=64$
6	$(6-11)^2=5^2=25$

The feature value 6 is near to the testing data point. So, it is predicted to be class value 1.

Feature Value	Original Class	Predicted Class
6	1	1
7	0	1
10	0	1
11	0	1

2.

Three values of class 1 of type O are predicted as false and one value of class X is predicted as true.

		Predicted Class	
		X	O
Actual Class	X	1	0
	O	3	0

True Positive(TP)=1

False Positive(FP)=3

True Negative(TN)=0

False Negative(FN)=0

Accuracy=(TP+FN)/(TP+FP+TN+FN)= $\frac{1}{4}$ =25%

Sensitivity=TP/(TP+FN)=1/(1+0)=100%

Specificity=(TN)/(TN+FP)=(0)/(0+3)=0