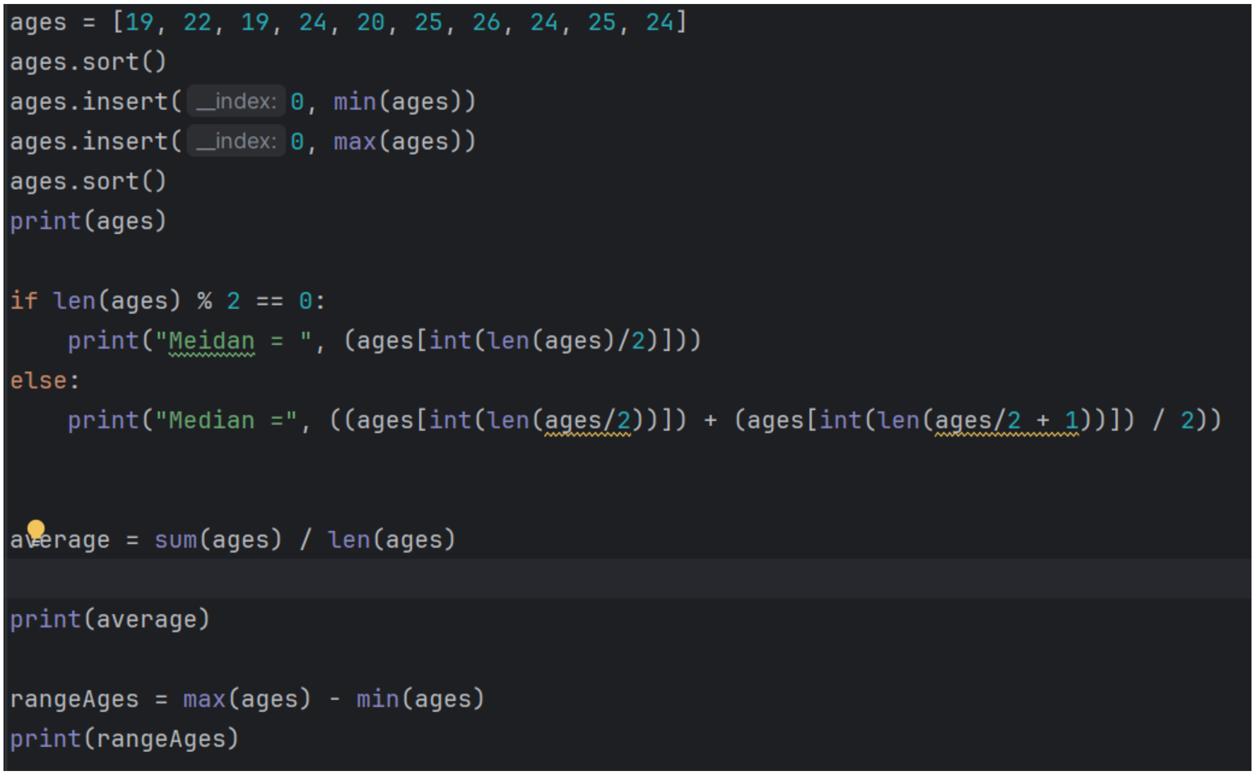
Amberly Stevenson

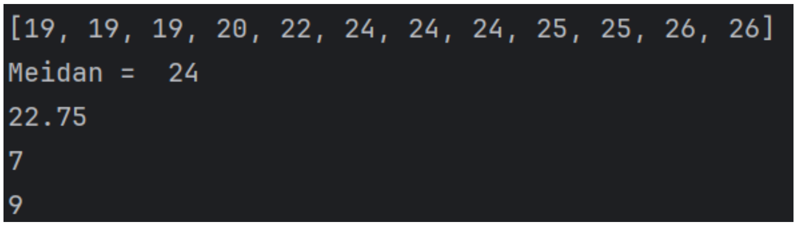
Machine learning: ICP 1

Code output

Q1:

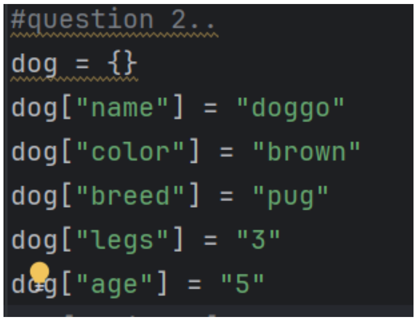


```
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
ages.sort()
ages.insert( _index: 0, min(ages))
ages.insert( _index: 0, max(ages))
ages.sort()
print(ages)

if len(ages) % 2 == 0:
    print("Meidan = ", (ages[int(len(ages)/2)]))
else:
    print("Median =", ((ages[int(len(ages/2))]) + (ages[int(len(ages/2 + 1))]) / 2))


average = sum(ages) / len(ages)

print(average)

rangeAges = max(ages) - min(ages)
print(rangeAges)
```

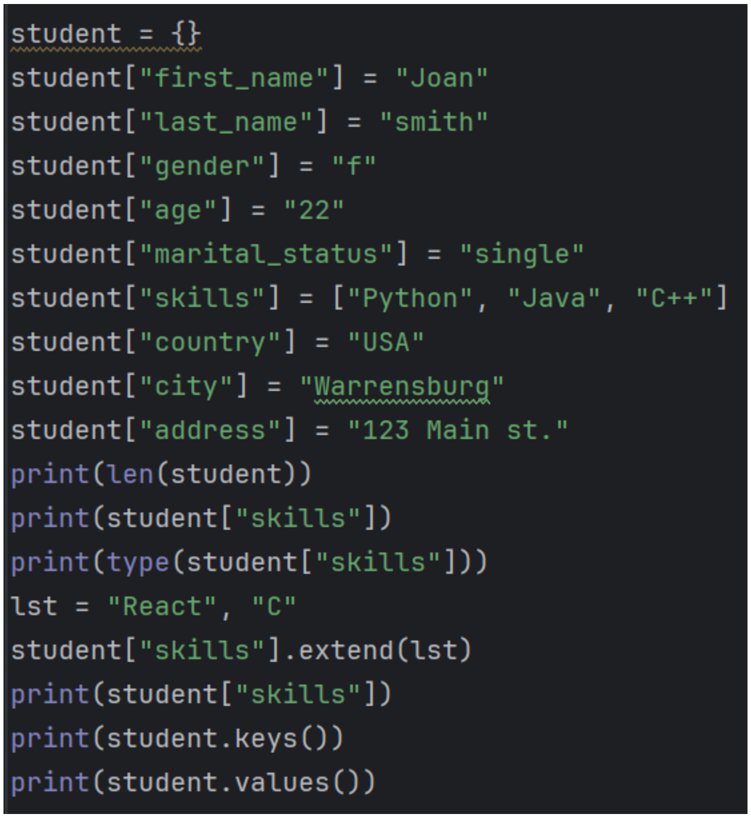


```
[19, 19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 26]
Meidan =  24
22.75
7
9
```

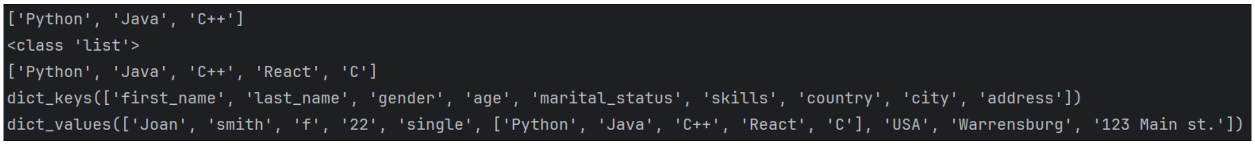
Q2:



```
#question 2..
dog = {}
dog["name"] = "doggo"
dog["color"] = "brown"
dog["breed"] = "pug"
dog["legs"] = "3"
dog["age"] = "5"
```

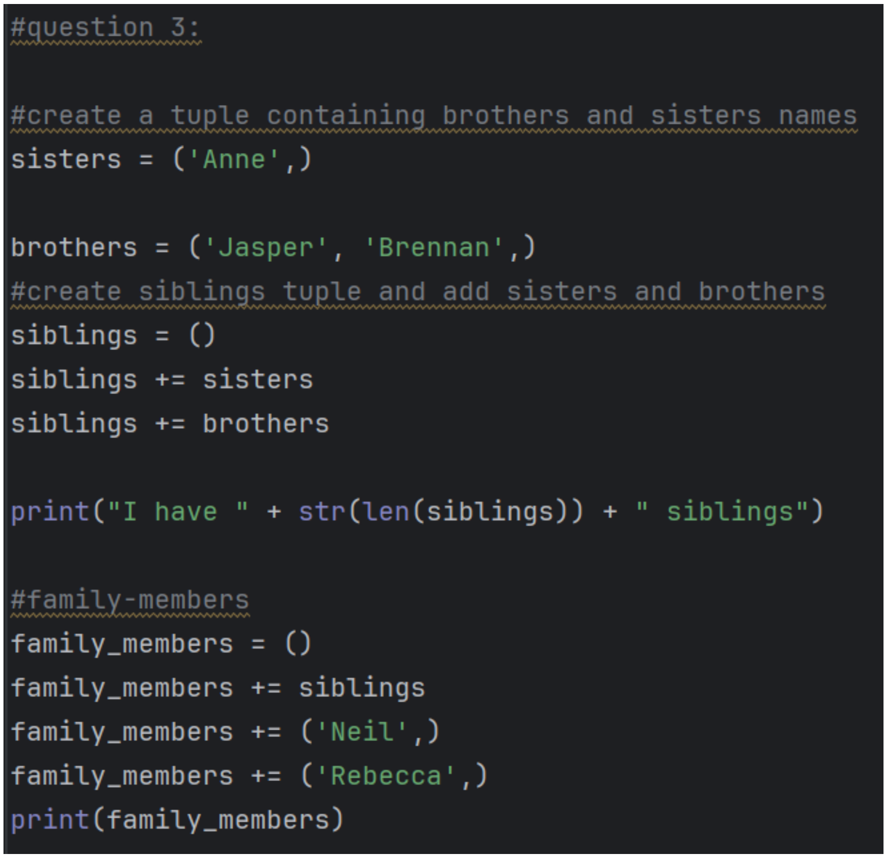


```python
student = {}
student["first_name"] = "Joan"
student["last_name"] = "smith"
student["gender"] = "f"
student["age"] = "22"
student["marital_status"] = "single"
student["skills"] = ["Python", "Java", "C++"]
student["country"] = "USA"
student["city"] = "Warrensburg"
student["address"] = "123 Main st."
print(len(student))
print(student["skills"])
print(type(student["skills"]))
lst = "React", "C"
student["skills"].extend(lst)
print(student["skills"])
print(student.keys())
print(student.values())
```

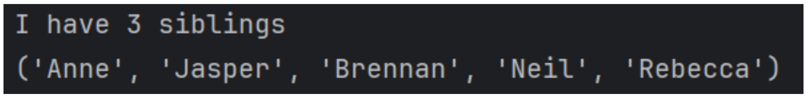


```
['Python', 'Java', 'C++']
<class 'list'>
['Python', 'Java', 'C++', 'React', 'C']
dict_keys(['first_name', 'last_name', 'gender', 'age', 'marital_status', 'skills', 'country', 'city', 'address'])
dict_values(['Joan', 'smith', 'f', '22', 'single', ['Python', 'Java', 'C++', 'React', 'C'], 'USA', 'Warrensburg', '123 Main st.'])
```

Q3:

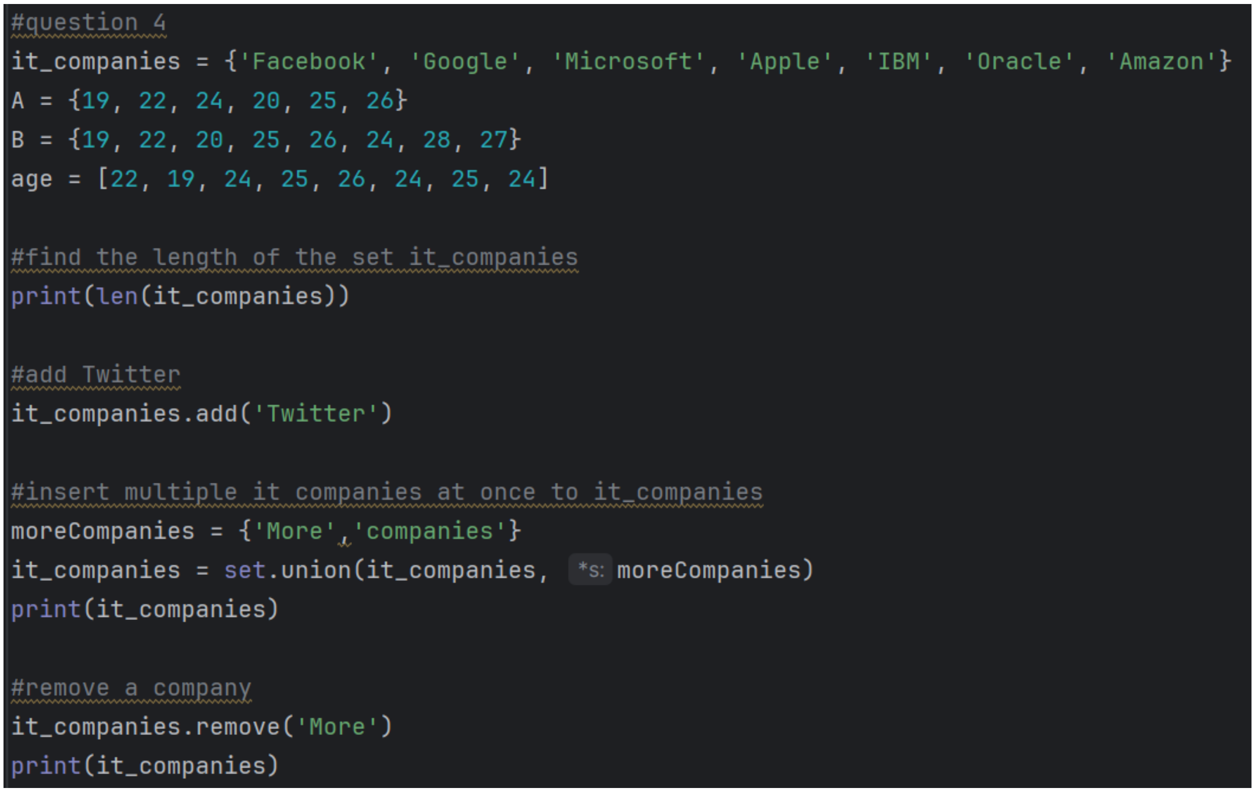


```
#question 3:

#create a tuple containing brothers and sisters names
sisters = ('Anne',)

brothers = ('Jasper', 'Brennan',)
#create siblings tuple and add sisters and brothers
siblings = ()
siblings += sisters
siblings += brothers

print("I have " + str(len(siblings)) + " siblings")

#family-members
family_members = ()
family_members += siblings
family_members += ('Neil',)
family_members += ('Rebecca',)
print(family_members)
```

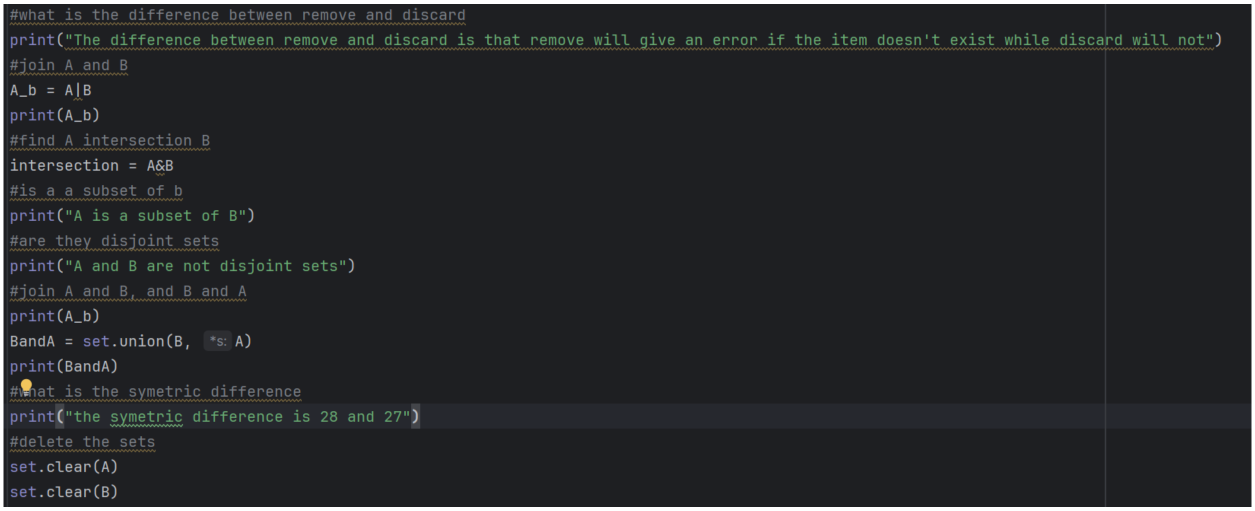


```
I have 3 siblings
('Anne', 'Jasper', 'Brennan', 'Neil', 'Rebecca')
```

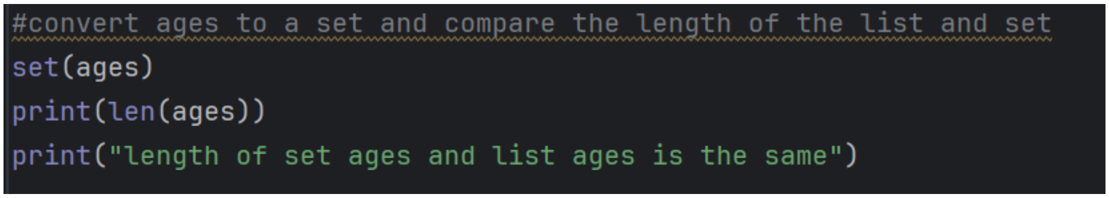
Q4:



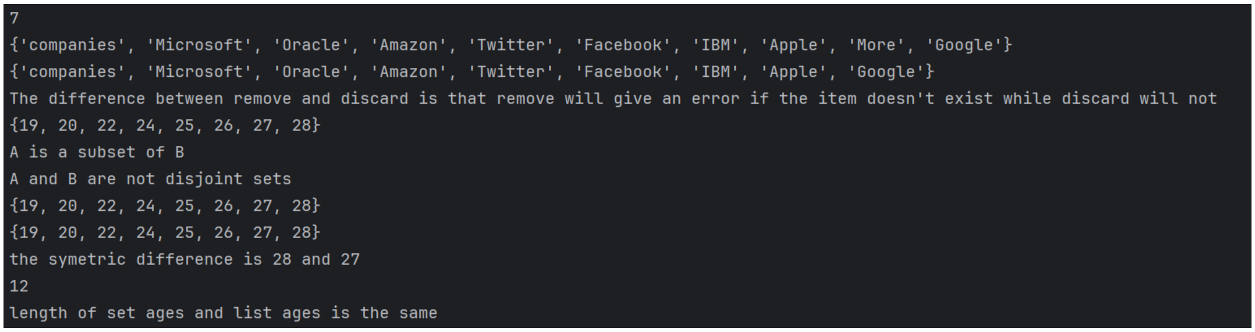
```
#question 4
it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
age = [22, 19, 24, 25, 26, 24, 25, 24]

#find the length of the set it_companies
print(len(it_companies))

#add Twitter
it_companies.add('Twitter')

#insert multiple it companies at once to it_companies
moreCompanies = {'More','companies'}
it_companies = set.union(it_companies, moreCompanies)
print(it_companies)

#remove a company
it_companies.remove('More')
print(it_companies)
```



```
#what is the difference between remove and discard
print("The difference between remove and discard is that remove will give an error if the item doesn't exist while discard will not")
#join A and B
A_b = A|B
print(A_b)
#find A intersection B
intersection = A&B
#is a a subset of b
print("A is a subset of B")
#are they disjoint sets
print("A and B are not disjoint sets")
#join A and B, and B and A
print(A_b)
BandA = set.union(B, A)
print(BandA)
#what is the symetric difference
print("the symetric difference is 28 and 27")
#delete the sets
set.clear(A)
set.clear(B)
```

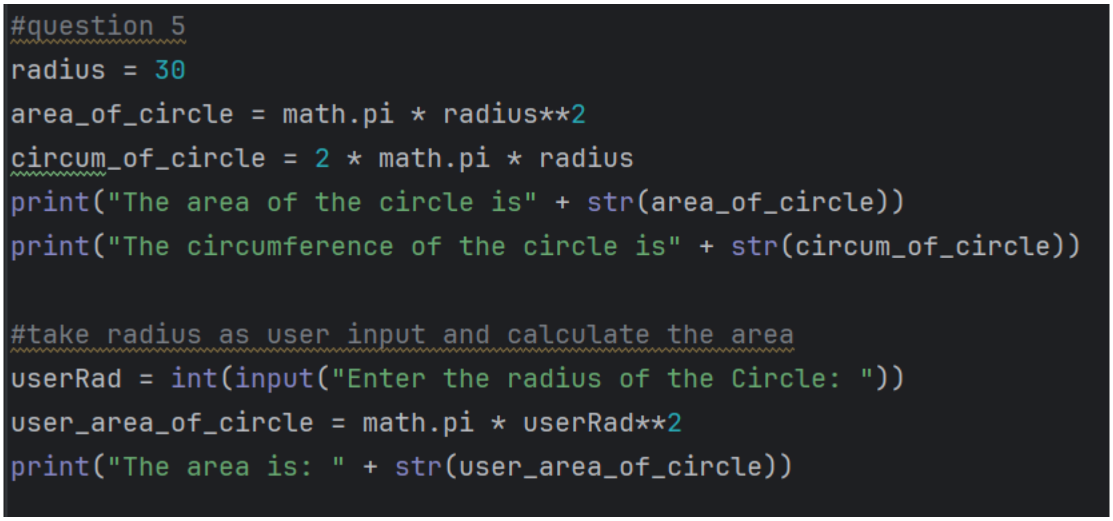


```
#convert ages to a set and compare the length of the list and set
set(ages)
print(len(ages))
print("length of set ages and list ages is the same")
```

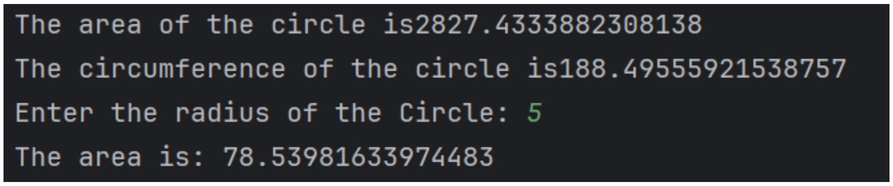


```
7
{'companies', 'Microsoft', 'Oracle', 'Amazon', 'Twitter', 'Facebook', 'IBM', 'Apple', 'More', 'Google'}
{'companies', 'Microsoft', 'Oracle', 'Amazon', 'Twitter', 'Facebook', 'IBM', 'Apple', 'Google'}
The difference between remove and discard is that remove will give an error if the item doesn't exist while discard will not
{19, 20, 22, 24, 25, 26, 27, 28}
A is a subset of B
A and B are not disjoint sets
{19, 20, 22, 24, 25, 26, 27, 28}
{19, 20, 22, 24, 25, 26, 27, 28}
the symetric difference is 28 and 27
12
length of set ages and list ages is the same
```

Q5:

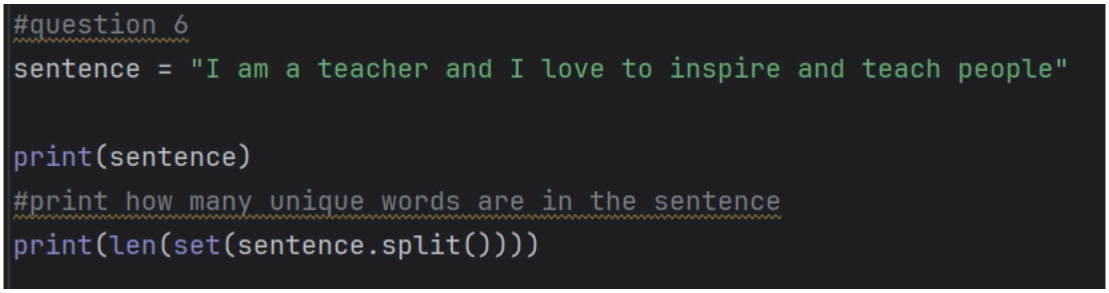


```
#question 5
radius = 30
area_of_circle = math.pi * radius**2
circum_of_circle = 2 * math.pi * radius
print("The area of the circle is" + str(area_of_circle))
print("The circumference of the circle is" + str(circum_of_circle))

#take radius as user input and calculate the area
userRad = int(input("Enter the radius of the Circle: "))
user_area_of_circle = math.pi * userRad**2
print("The area is: " + str(user_area_of_circle))
```

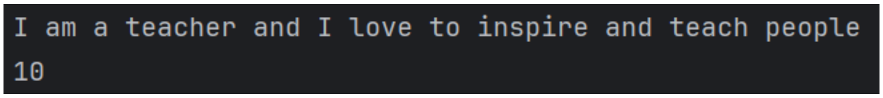


```
The area of the circle is2827.4333882308138
The circumference of the circle is188.49555921538757
Enter the radius of the Circle: 5
The area is: 78.53981633974483
```

Q6:

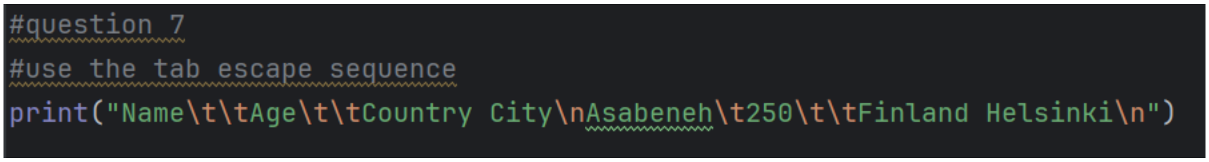


```
#question 6
sentence = "I am a teacher and I love to inspire and teach people"

print(sentence)
#print how many unique words are in the sentence
print(len(set(sentence.split())))
```

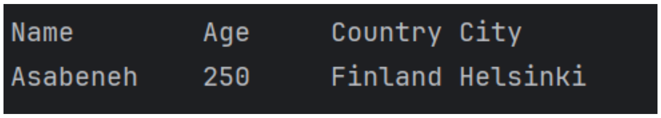


```
I am a teacher and I love to inspire and teach people
10
```

Q7:

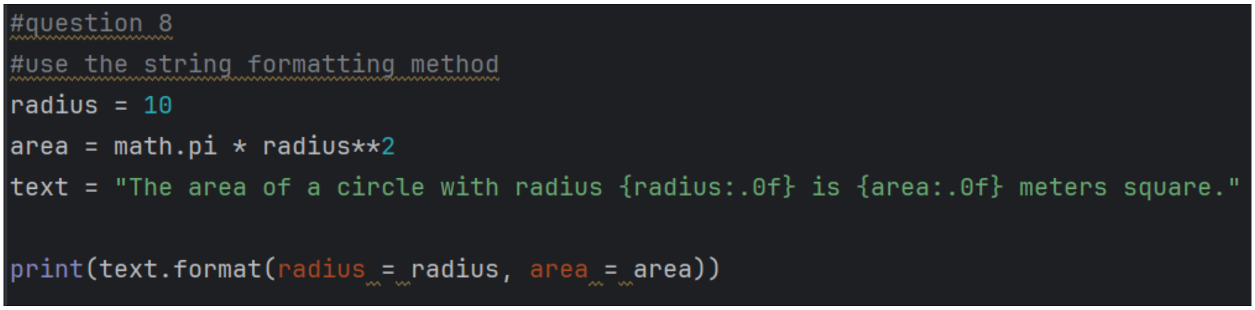


```
#question 7
#use the tab escape sequence
print("Name\t\tAge\t\tCountry City\nAsabeneh\t250\t\tFinland Helsinki\n")
```

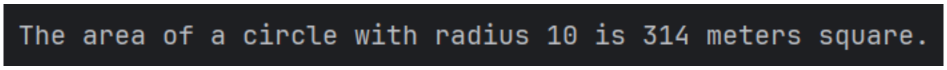


```
Name        Age     Country City
Asabeneh    250     Finland Helsinki
```

Q8:

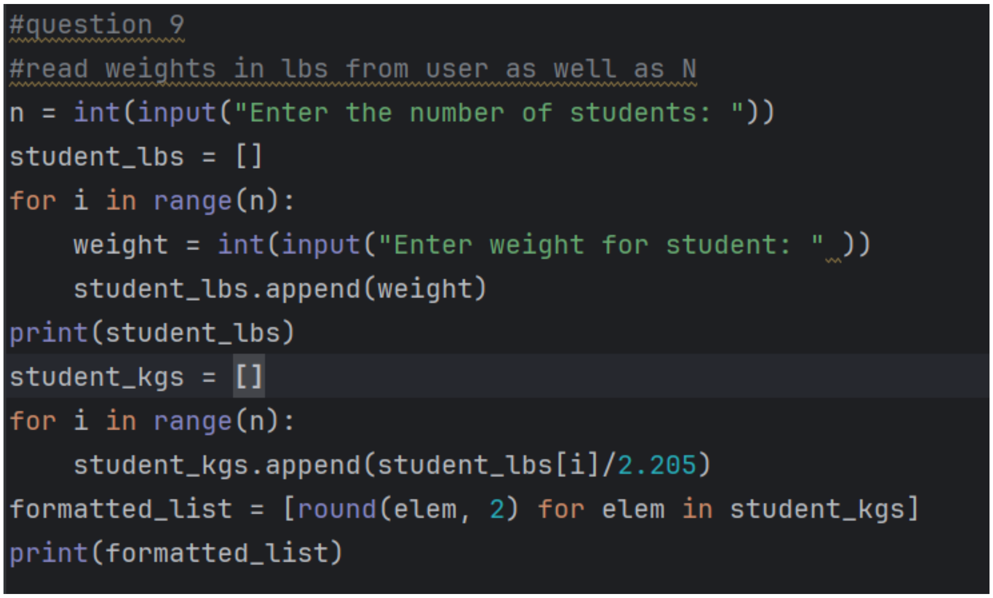


```
#question 8
#use the string formatting method
radius = 10
area = math.pi * radius**2
text = "The area of a circle with radius {radius:.0f} is {area:.0f} meters square."

print(text.format(radius = radius, area = area))
```

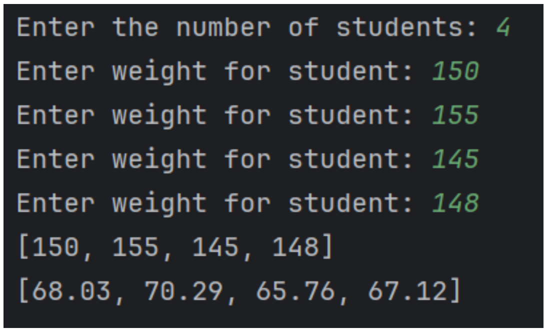


```
 The area of a circle with radius 10 is 314 meters square.
```

Q9:



```
#question 9
#read weights in lbs from user as well as N
n = int(input("Enter the number of students: "))
student_lbs = []
for i in range(n):
    weight = int(input("Enter weight for student: " ))
    student_lbs.append(weight)
print(student_lbs)
student_kgs = []
for i in range(n):
    student_kgs.append(student_lbs[i]/2.205)
formatted_list = [round(elem, 2) for elem in student_kgs]
print(formatted_list)
```



```
Enter the number of students: 4
Enter weight for student: 150
Enter weight for student: 155
Enter weight for student: 145
Enter weight for student: 148
[150, 155, 145, 148]
[68.03, 70.29, 65.76, 67.12]
```

Video link: https://youtu.be/b2iawm0LWiw