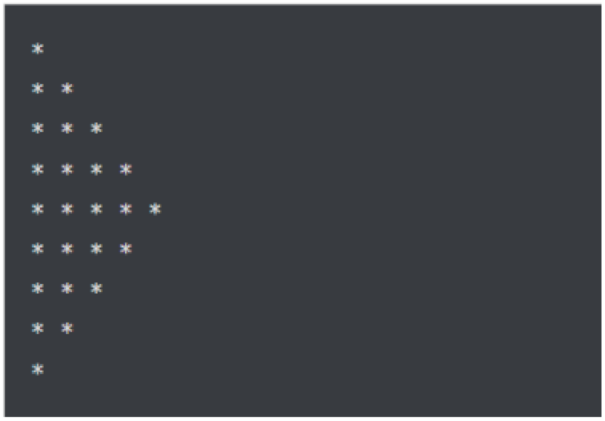
Spring 2023: CS5710 – Machine Learning

In-Class Programming Assignment-2

GitHub Link - https://github.com/raimukul/MachineLearning_Assignments

Video link-
https://drive.google.com/file/d/1qp0TJYqW4no51K0m6qG21c5vklvUhooA/view?usp=share_link

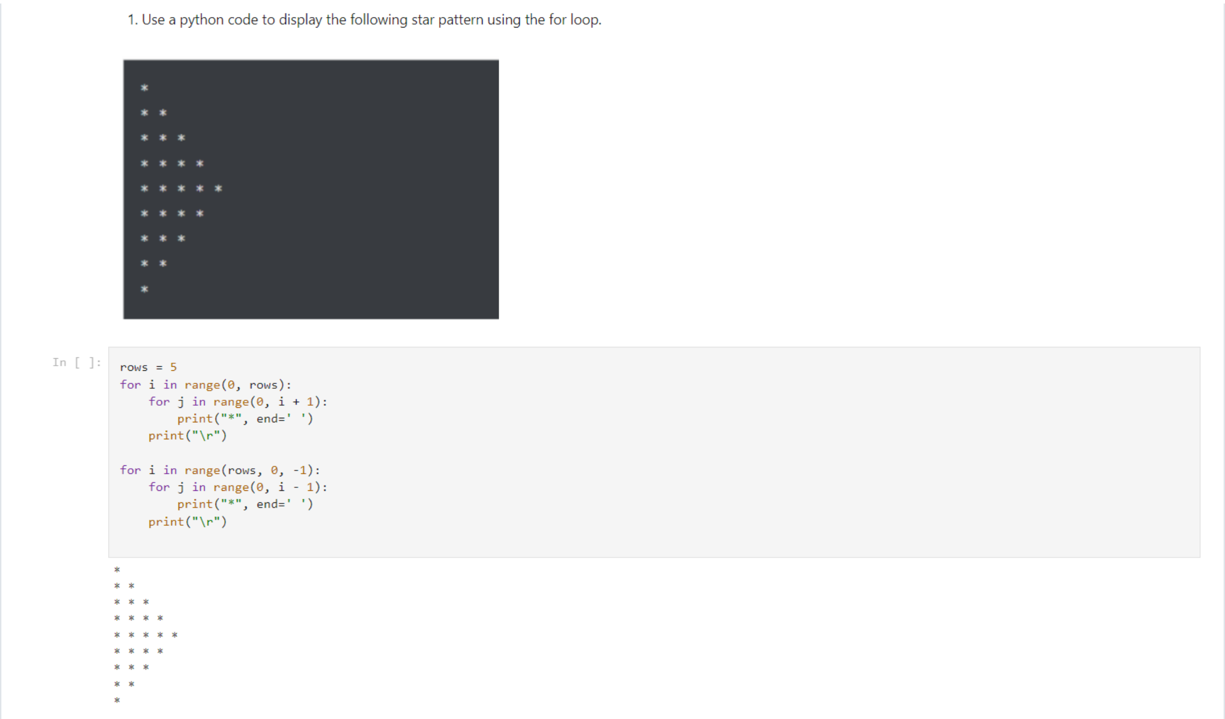
1. Use a python code to display the following star pattern using the for loop.



```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

```
#Define the number of rows.
rows = 5
#Use range in rows by for loop
for i in range(0, rows):
#increase the rows upto 5 and print * on completion of one row
    for j in range(0, i + 1):
        print("*", end=' ')
    print("\r")

#decrease the rows upto 1
for i in range(rows, 0, -1):
    for j in range(0, i - 1):
        print("*", end=' ')
    print("\r")
```



1. Use a python code to display the following star pattern using the for loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

In [ ]:

```
rows = 5
for i in range(0, rows):
    for j in range(0, i + 1):
        print("*", end=' ')
    print("\r")

for i in range(rows, 0, -1):
    for j in range(0, i - 1):
        print("*", end=' ')
    print("\r")
```

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

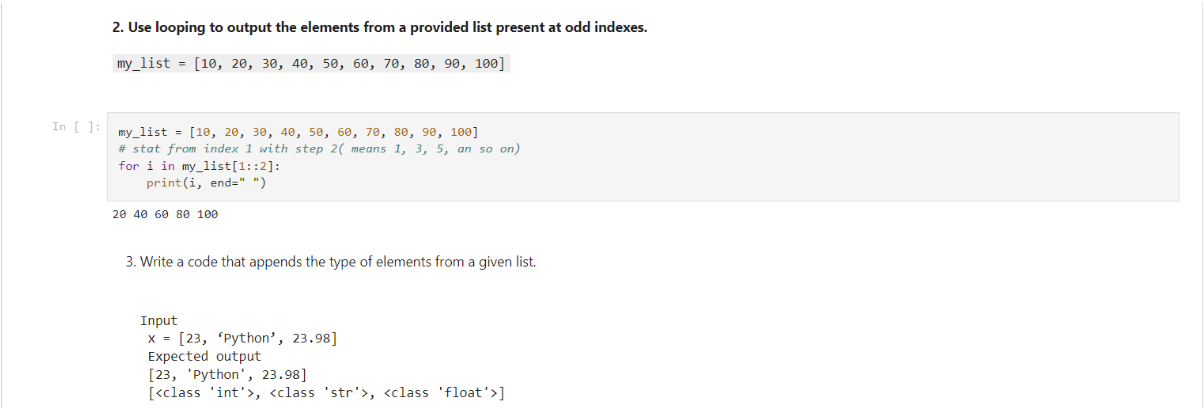
2. Use looping to output the elements from a provided list present at odd indexes.

```
#We define the list
my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

# start from index 1 with step 2( means 1, 3, 5, and so on)

for i in my_list[1::2]:
   print(i, end=" ")

20 40 60 80 100
```



2. Use looping to output the elements from a provided list present at odd indexes.

`my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]`

In [ ]:

```
my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# stat from index 1 with step 2( means 1, 3, 5, an so on)
for i in my_list[1::2]:
    print(i, end=" ")
```

```
20 40 60 80 100
```

3. Write a code that appends the type of elements from a given list.

```
Input
 x = [23, ‘Python’, 23.98]
 Expected output
 [23, 'Python', 23.98]
 [<class 'int'>, <class 'str'>, <class 'float'>]
```

3. Write a code that appends the type of elements from a given list

Input

x = [23, ‘Python’, 23.98]

Expected output

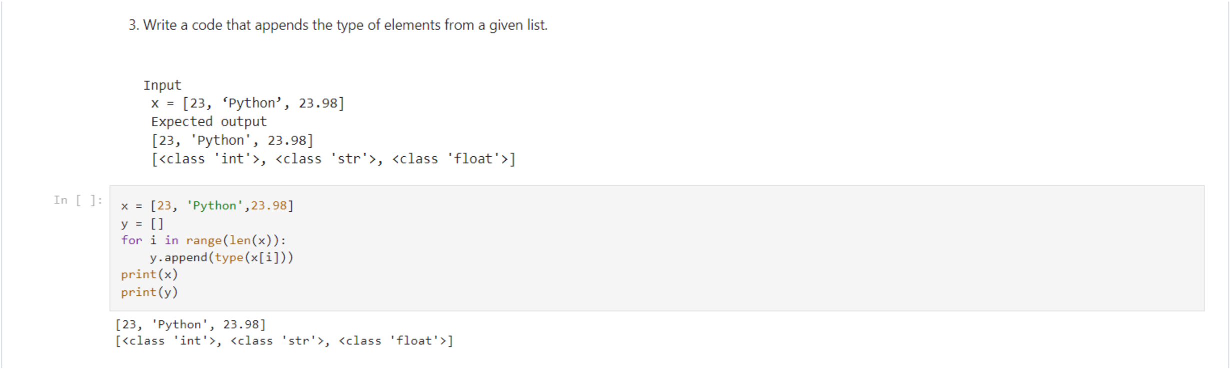
[23, 'Python', 23.98]

[<class 'int'>, <class 'str'>, <class 'float'>]

In [ ]:

```
x = [23, 'Python',23.98]
#create an empty list y.
y = []
for i in range(len(x)):
#appending the data types in list y
   y.append(type(x[i]))
print(x)
print(y)
```

```
[23, 'Python', 23.98]
[<class 'int'>, <class 'str'>, <class 'float'>]
```



3. Write a code that appends the type of elements from a given list.

```
Input
 x = [23, ‘Python’, 23.98]
 Expected output
 [23, 'Python', 23.98]
 [<class 'int'>, <class 'str'>, <class 'float'>]
```

In [ ]:

```
x = [23, 'Python',23.98]
y = []
for i in range(len(x)):
    y.append(type(x[i]))
print(x)
print(y)
```

```
[23, 'Python', 23.98]
[<class 'int'>, <class 'str'>, <class 'float'>]
```

4. Write a function that takes a list and returns a new list with unique items of the first list.

Sample List: [1,2,3,3,3,3,4,5]

Unique List: [1, 2, 3, 4, 5]

In [ ]:

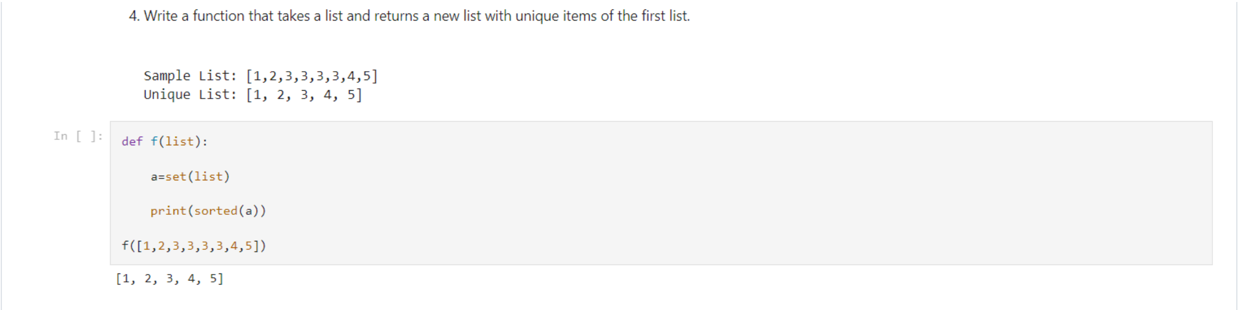
```
def f(list):

    a=set(list)

    print(sorted(a))

f([1,2,3,3,3,3,4,5])
```

```
[1, 2, 3, 4, 5]
```



4. Write a function that takes a list and returns a new list with unique items of the first list.

```
Sample List: [1,2,3,3,3,3,4,5]
Unique List: [1, 2, 3, 4, 5]
```

In [ ]:

```
def f(list):

    a=set(list)

    print(sorted(a))

f([1,2,3,3,3,3,4,5])
```

```
[1, 2, 3, 4, 5]
```

5. Write a function that accepts a string and calculate the number of upper-case letters and lower-case

letters.

Input String: 'The quick Brow Fox'

Expected Output:

No. of Upper-case characters: 3

No. of Lower-case Characters: 12

In [ ]:

```
def string_test(s):
    d={"upperCase":0, "lowerCase":0}
    for c in s:
        if c.isupper():
            d["upperCase"]+=1
        elif c.islower():
            d["lowerCase"]+=1
        else:
            pass
    print ("Original String : ", s)
    print ("No. of Upper-case characters : ", d["upperCase"])
    print ("No. of Lower-case Characters : ", d["lowerCase"])
string_test('The quick Brow Fox')
```

```
Original String :  The quick Brow Fox
No. of Upper-case characters :  3
No. of Lower-case Characters :  12
```

5. Write a function that accepts a string and calculate the number of upper-case letters and lower-case

letters.

```
Input String: 'The quick Brow Fox'
Expected Output:
No. of Upper-case characters: 3
No. of Lower-case Characters: 12
```

In [ ]:

```
def string_test(s):
    d={"upperCase":0, "lowerCase":0}
    for c in s:
        if c.isupper():
           d["upperCase"]+=1
        elif c.islower():
           d["lowerCase"]+=1
        else:
           pass
    print ("Original String : ", s)
    print ("No. of Upper-case characters : ", d["upperCase"])
    print ("No. of Lower-case Characters : ", d["lowerCase"])
string_test('The quick Brow Fox')
```

```
Original String :  The quick Brow Fox
No. of Upper-case characters :  3
No. of Lower-case Characters :  12
```