Assignment - 2

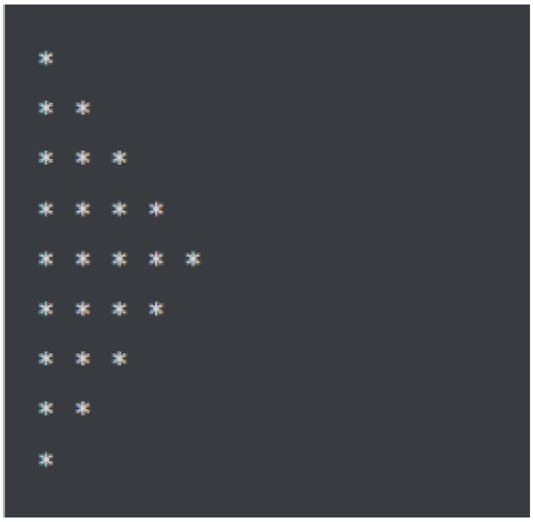
Machine Learning (CS 5710) CRN: 22002

Vinay Kumar Camarushi
700740428

GitHub Link: https://github.com/VXC04280/In_Class_Programming_Assignment_2

Question – 1)

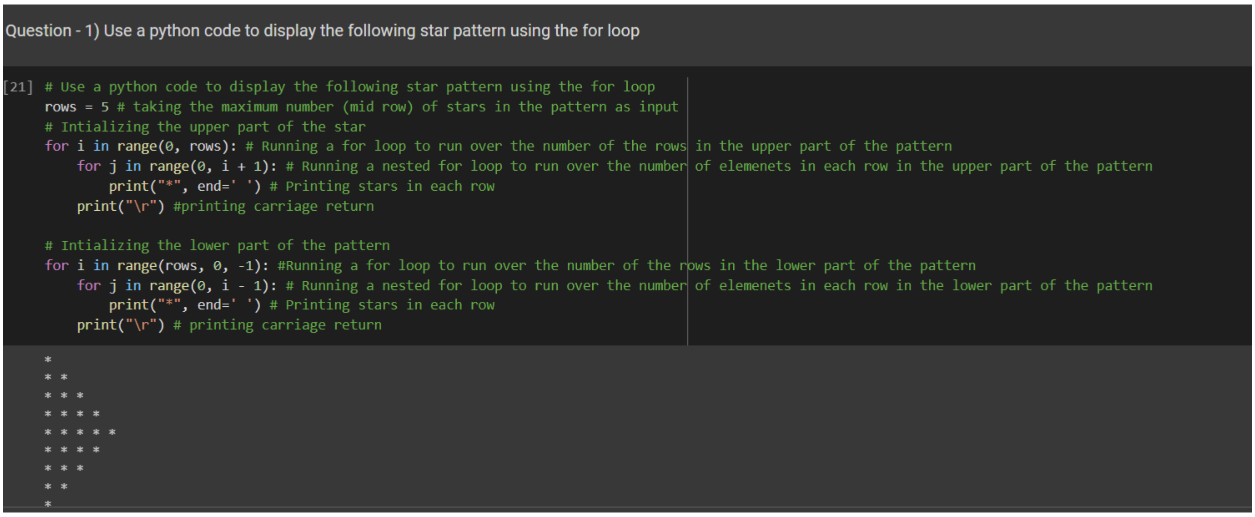
Use a python code to display the following star pattern using the for loop



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

Solution :

- Here we used to nested for loops to print the given star pattern.
- The first nested loop is used to print the upper part of the star pattern and the lower for loop is used to print the lower part of the string.
- The first for loop in both the parts are used to loop over the number of rows in the pattern.
- The Second for loop is used to loop over the number of elements of the rows based on the iterable of the first for loop.



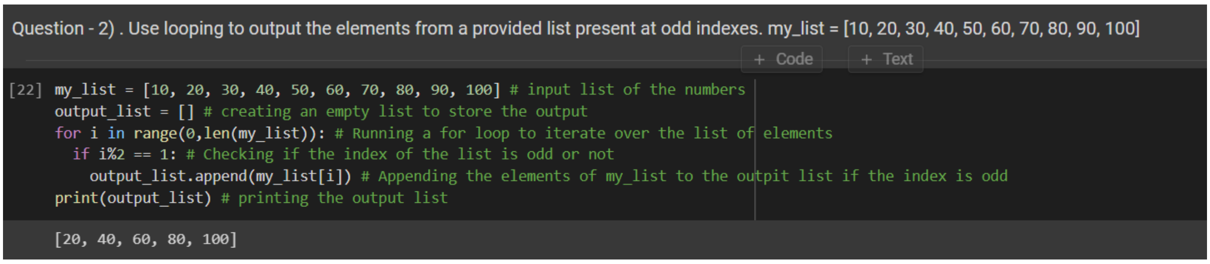
Question - 1) Use a python code to display the following star pattern using the for loop

```
[21] # Use a python code to display the following star pattern using the for loop
     rows = 5 # taking the maximum number (mid row) of stars in the pattern as input
     # Intializing the upper part of the star
     for i in range(0, rows): # Running a for loop to run over the number of the rows in the upper part of the pattern
         for j in range(0, i + 1): # Running a nested for loop to run over the number of elemenets in each row in the upper part of the pattern
             print("*", end=' ') # Printing stars in each row
         print("\r") #printing carriage return

     # Intializing the lower part of the pattern
     for i in range(rows, 0, -1): #Running a for loop to run over the number of the rows in the lower part of the pattern
         for j in range(0, i - 1): # Running a nested for loop to run over the number of elemenets in each row in the lower part of the pattern
             print("*", end=' ') # Printing stars in each row
         print("\r") # printing carriage return
```

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

Question – 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]

- Here the input list is stored in my_list
- Created an empty list for storing the output
- We ran a for loop to check if the index is even or odd.
- If the index is odd, we appended the list elements to the output_list
- At last we printed the output_list



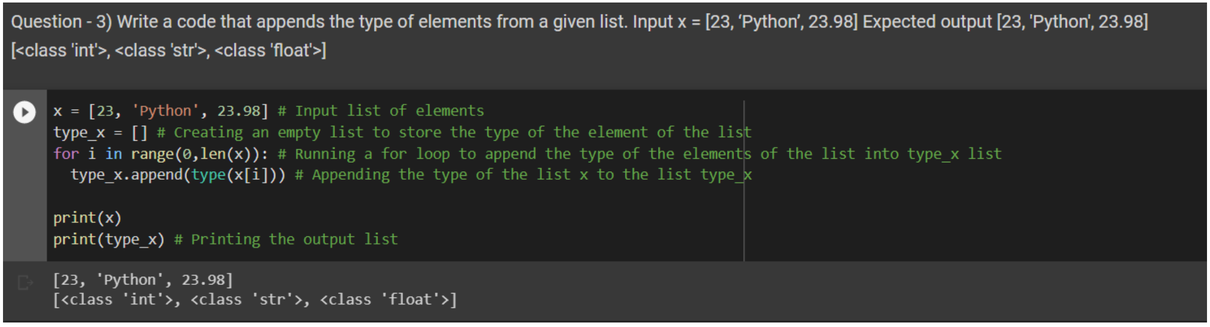
Question - 2) . Use looping to output the elements from a provided list present at odd indexes. my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

```
[22] my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] # input list of the numbers
     output_list = [] # creating an empty list to store the output
     for i in range(0,len(my_list)): # Running a for loop to iterate over the list of elements
       if i%2 == 1: # Checking if the index of the list is odd or not
         output_list.append(my_list[i]) # Appending the elements of my_list to the outpit list if the index is odd
     print(output_list) # printing the output list
```

```
[20, 40, 60, 80, 100]
```

Question – 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<’int’>]

- Here the input list is stored in x
- Created an empty list for storing the output
- We ran a for loop to iterate over the loop to get the type of the index.
- We appended the type of the list elements to the output_list
- At last we printed the output_list



Question - 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'>]

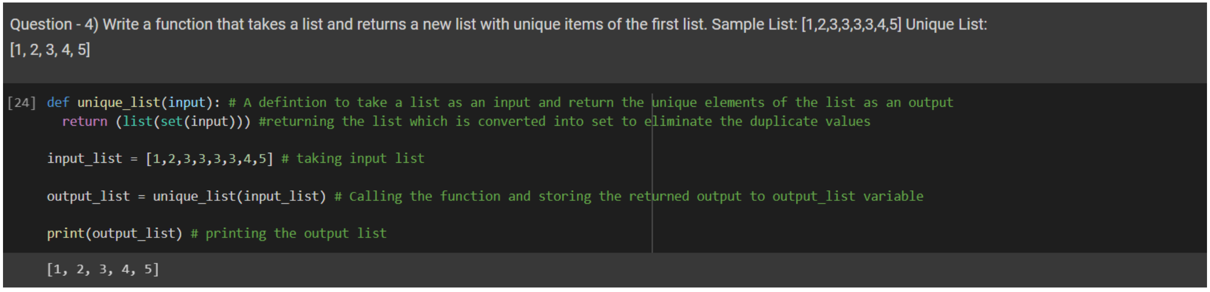
```
x = [23, 'Python', 23.98] # Input list of elements
type_x = [] # Creating an empty list to store the type of the element of the list
for i in range(0,len(x)): # Running a for loop to append the type of the elements of the list into type_x list
  type_x.append(type(x[i])) # Appending the type of the list x to the list type_x

print(x)
print(type_x) # Printing the output list
```

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

Question – 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]

- Here the input list is stored in input_list
- Created a function to get the unique elements of the list.
- In the method, we convert the input list to a set and then type cast the set as a list again.
- We return the list from the given method.
- When the function is called the output is stored in the variable output_list
- Output_list is printed.



Question - 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]

```
def unique_list(input): # A defintion to take a list as an input and return the unique elements of the list as an output
  return (list(set(input))) #returning the list which is converted into set to eliminate the duplicate values

input_list = [1,2,3,3,3,3,4,5] # taking input list

output_list = unique_list(input_list) # Calling the function and storing the returned output to output_list variable

print(output_list) # printing the output list
```

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

Question – 5)

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

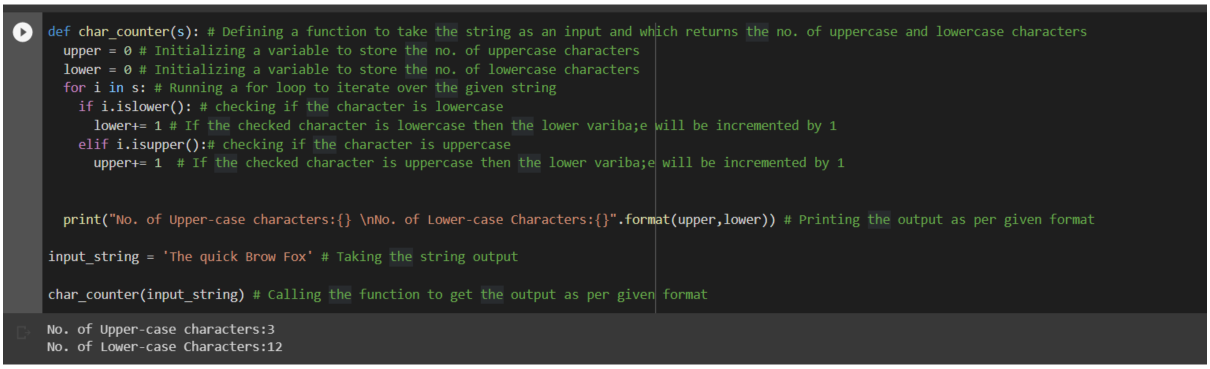
Input String: 'The quick Brow Fox'

Expected Output:

No. of Upper-case characters: 3

No. of Lower-case Characters: 12

- Here the input string is stored in input_string and pass it to the definition to get the output.
- Created a function to get the no. of uppercase and lowercase letters
- In the method, we iterate over the string and check if the character is lowercase or uppercase.
- If the character is uppercase, then the upper counter will be incremented.
- If the character is lowercase, then the lower counter will be incremented.
- We print the statements as per the given question.



```
def char_counter(s): # Defining a function to take the string as an input and which returns the no. of uppercase and lowercase characters
  upper = 0 # Initializing a variable to store the no. of uppercase characters
  lower = 0 # Initializing a variable to store the no. of lowercase characters
  for i in s: # Running a for loop to iterate over the given string
    if i.islower(): # checking if the character is lowercase
      lower+= 1 # If the checked character is lowercase then the lower variba;e will be incremented by 1
    elif i.isupper():# checking if the character is uppercase
      upper+= 1  # If the checked character is uppercase then the lower variba;e will be incremented by 1


  print("No. of Upper-case characters:{} \nNo. of Lower-case Characters:{}".format(upper,lower)) # Printing the output as per given format

input_string = 'The quick Brow Fox' # Taking the string output

char_counter(input_string) # Calling the function to get the output as per given format
```

```
No. of Upper-case characters:3
No. of Lower-case Characters:12
```