CS 5710 Machine Learning

In-Class Programming Assignment-2

Name: Aravinda Krishna Gorantla

ID: 700741775

GitHub Link: https://github.com/gakrish5/MachineLearning/tree/main/Assignment%202

Question 1:

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

Source code:

```
rows = 5 #gives the max. stars in a line

for i in range(0, rows+1):
    #prints required no. of stars in each line by incrementing the value of i
    print("* " * i)

for j in range(rows-1, 0, -1):
    #prints required no. of stars in each line by decrementing the value of j as step value is given by -1
    print("* " * j)
```

Output:

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

Explanation:

Here in the code, I have taken a variable $rows$, which has the value of maximum number of stars in a line. I used to 2 $for\ loops$, one to print stars in increasing order until it reaches the value of $rows$ and then second loop prints stars in decreasing order.

Question 2:

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

$my\_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$

Source code:

```
my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] #input list
odd_elements = [] #empty list

for i in range(0, len(my_list)):
    if i%2 != 0:  #checking the odd index
        odd_elements.append(my_list[i]) #appending the elements to a list
        print(my_list[i]) #printing individual elements

print("\nOdd indexes elements list:", odd_elements) #printing the list of odd ondex elements
```

Output:

```
20
40
60
80
100

Odd indexes elements list: [20, 40, 60, 80, 100]
```

Explanation:

Here in the code, I have declared and initialized the given list: my_list and created an empty list: $odd\_elements$.

I have traversed through the indexes of the my_list using $for\ loop$ and checked the odd index using $if\ statement$.

Once the odd index is found, the value of it is printed and is appended to $odd\_elements$ list. The $odd\_elements$ list which is the list of odd index elements is also printed.

Question 3:

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

Input

$$x = [23, 'Python', 23.98]$$

Expected output

$$[23, 'Python', 23.98]$$

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

Source code:

```
x = [23, "Python", 23.98] #input list
y = [] #empty list

for i in x:
    y.append(type(i)) #appending the type of elements in x list to y list

print(x) #printing x list
print(y) #printing y list
```

Output:

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

Explanation:

Here in the code, I have declared and initialized the given list: x and created an empty list y.

I have traversed through the elements of the x list using $for\ loop$ and then the $type$ of individual element is appended to the y list.

Then the both the lists are 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]

Source code:

```
#function to return unique items of input list
def unique(list1):
    list2 = list(set(list1)) #set removes duplicates
    return list2

list1 = [1,2,3,3,3,3,4,5] #input list
print("unique list:", unique(list1)) #function call
```

Output:

```
unique list: [1, 2, 3, 4, 5]
```

Explanation:

Here in the code, I have created a function named *unique*, which takes an argument specifically a list and returns a unique elements list.

To get the unique elements, list is converted to set and then back to list, as set doesn't allow duplicates.

I have printed the unique list finally using print function, where in I have made function call for *unique* function with a list.

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

Source code:

```
def lettercase(str1):
    up = 0; low = 0
    for i in str1:
        if i.isupper(): #checking upper case
            up += 1
        elif i == ' ': #checking space character
            continue
        else: #other than upper case and space char => lower case
            low += 1

    #printing required output
    print("No. of Upper-case characters: {} \nNo. of Lower-case Characters: {}".format(up, low))


string ='The quick Brow Fox' #Input string
lettercase(string) #function call
```

Output:

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

Explanation:

Here in the code, I have created a function named *lettercase*, which takes an argument specifically a string and returns a required output string.

In the function, 2 counter variables are used one for the upper case and the other for lower case. Using *for loop*, I have traversed each character of the string and checked the letter case using *if statement*. I have incremented the values of the counter variables as and when the case is found.

Finally, when traversal of the string is done, the required output is printed.

- - - - End - - - -