**Summer 2024: CS 5710 –Machine Learning**

**Programming Assignment-2**

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

# Define the number of rows

rows = 5

# Display the upper half of the pattern

for i in range(1, rows + 1):

for j in range(1, i + 1):

print("\*", end=" ")

print()

# Display the lower half of the pattern

for i in range(rows - 1, 0, -1):

for j in range(1, i + 1):

print("\*", end=" ")

print()

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]

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

# Loop through the indices of the list and output the elements at odd indices

for i in range(1, len(my\_list), 2):

print(my\_list[i])

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] [, , ]

x = [23, 'Python', 23.98]

# List to store the types of elements

types\_list = []

# Iterate over the elements of the list

for element in x:

# Append the type of each element to the types\_list

types\_list.append(type(element))

# Print the original list

print(x)

# Print the list containing the types of elements

print(types\_list)

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\_list):

# Create an empty list to store unique elements

unique\_items = []

# Iterate over the input list

for item in input\_list:

# Check if the item is not already in the unique list

if item not in unique\_items:

# Add the item to the unique list

unique\_items.append(item)

# Return the list with unique items

return unique\_items

# Sample list

sample\_list = [1, 2, 3, 3, 3, 3, 4, 5]

# Call the function and print the unique list

print("Sample List:", sample\_list)

print("Unique List:", unique\_list(sample\_list))

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

def count\_case\_characters(input\_string):

# Initialize counters for upper-case and lower-case characters

upper\_count = 0

lower\_count = 0

# Iterate through each character in the input string

for char in input\_string:

# Check if the character is an upper-case letter

if char.isupper():

upper\_count += 1

# Check if the character is a lower-case letter

elif char.islower():

lower\_count += 1

# Return the counts of upper-case and lower-case characters

return upper\_count, lower\_count

# Input string

input\_string = 'The quick Brow Fox'

# Call the function and print the counts

upper\_count, lower\_count = count\_case\_characters(input\_string)

print("No. of Upper-case characters:", upper\_count)

print("No. of Lower-case Characters:", lower\_count)