**Name:** Ahmad Hashim Farooq

**Reg No:** FA21-BCS-138

**TASK 1**

**1. Define Linear and non-Linear data structures with examples & implementation.**

Certainly! Here's an explanation of linear and non-linear data structures along with examples:

**Linear Data Structures:**

Linear data structures are those in which elements are arranged in a sequential manner, with each element connected to its previous and next element. These structures follow a linear order of arrangement.

**Arrays:**

Arrays are collections of elements stored at contiguous memory locations.

Elements in an array are accessed using their indices.

Example: [1, 2, 3, 4, 5]

**Linked Lists:**

Linked lists are made up of nodes where each node contains a data field and a reference (link) to the next node in the sequence.

There are different types of linked lists such as singly linked lists, doubly linked lists, and circular linked lists.

Example: Singly Linked List: 1 -> 2 -> 3 -> 4 -> 5

**Stacks:**

Stacks are a collection of elements with last in, first out (LIFO) access.

Elements are added and removed from the top of the stack.

Example: [5, 4, 3, 2, 1] (When pushing elements onto the stack and popping them off)

**Queues:**

Queues are a collection of elements with first in, first out (FIFO) access.

Elements are added at the rear and removed from the front of the queue.

Example: [1, 2, 3, 4, 5] (When enqueueing elements into the queue and dequeuing them out)

**Non-Linear Data Structures:**

Non-linear data structures are those in which elements are not arranged in a sequential manner. These structures can have elements connected in multiple directions or hierarchically.

**Trees:**

Trees are hierarchical data structures consisting of nodes connected by edges.

Each node has a parent node and zero or more child nodes.

Examples include binary trees, binary search trees, AVL trees, etc.

**Graphs:**

Graphs are collections of nodes (vertices) and edges that connect these nodes.

Graphs can be directed or undirected, and edges can have weights or be unweighted.

**Heaps:**

Heaps are specialized binary trees used in priority queue implementations.

They satisfy the heap property where the key at each node is greater than or equal to the keys of its children.

Examples include max heaps and min heaps.

**2. Type Casting**

Type casting in Python refers to the process of converting a variable from one data type to another. Python provides built-in functions for type casting, making it easy to convert between different data types.

**Examples:**

x = 5.6

y = int(x)

print(y)  # Output: 5

x = 10

y = float(x)

print(y)  # Output: 10.0

**3. For/While Loop**

For loop:

The for loop iterates over a sequence (such as a list, tuple, string, or range) or any iterable object. It executes a block of code for each item in the sequence.

Iterating over a list

fruits = ["apple", "banana", "cherry"]

for fruit in fruits:

    print(fruit)

Iterating over a range

for num in range(1, 5):

    print(num)

**While loop:**

The while loop repeats a block of code as long as a specified condition is true.

**Using while loop to count down from 5 to 1.**

num = 5

while num > 0:

    print(num)

    num -= 1

**4. Strings**

Strings in Python are sequences of characters, enclosed within either single quotes (' '), double quotes (" "), or triple quotes (''' ''' or """ """). Python treats single quotes the same as double quotes, and there's no functional difference between them. Triple quotes are used for multiline strings or for strings that contain both single and double quotes.

Single line string

str1 = "Hello, World!"

Multi-line string using triple quotes.

str2 = '''This is a

multi-line

string'''

**5. Implement Range Function and write at least 3 Examples.**

for i in range(5):

    print(i, end=" ")  # Output: 0 1 2 3 4

for i in range(2, 8):

    print(i, end=" ")  # Output: 2 3 4 5 6 7

for i in range(1, 10, 2):

    print(i, end=" ")  # Output: 1 3 5 7 9

**6. Print Odd Numbers from 49 to 35 using While loop.**

i=49*;*

while i>34:

    if(i%2!=0):

        print(i)

    i=i-1