Estd. 2000

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

MUST KNOW CONCEPTS

MKC

IT

2021-22

Course Code & Course Name : 19GES32-Data Structures using Python

Year/Sem/Sec : II / IV / A ,B &C

S. No.	Term	Notation (Symbol)	Concept/Definition/Meaning/Units/Equation/ Expression	Units
UNIT-I: LINEAR DATA STRUCTURES				
1.	Data structure	-	Data structure defines a way of organizing all data items	-
2.	Static data structures	-	A data structure formed when the number of data items are known in advance is referred as static data structure or fixed size data structure.	-
3.	Dynamic data structures	-	A data structure formed when the number of data items are not known in advance is known as dynamic data structure or variable size data structure.	-
4.	Types of data structures	-	• Linear data structure • Non-linear data structure	-
5.	Linear data structure	-	Linear data structures are data structures having a linear relationship between its adjacent elements	-
6.	Non-linear data structure	-	Non-linear data structures are data structures that don’t have a linear relationship between its adjacent elements but have a hierarchical relationship between the elements.	-
7.	Abstract Data Type(ADT)	-	An Abstract Data Type is a set of operations for which the implementation of the data structure is not specified anywhere in the program	-
8.	Array	-	Array is a collection of variables belongings to the same data type. Can store group of data of same data type in an array	-
9.	Applications of array	-	1. It stores elements of same data type. 2. Arrays can perform matrix operation. 3. Code optimization. 4. Array can be used in recursive function.	-
10.	List	-	A sequence of values.	-
11.	Element	-	One of the values in a list (or other sequence), also called items.	-
12.	Nested List	-	A list that is an element of another list.	-
13.	Tuple	-	An immutable sequence of elements.	-
14.	Interactive mode	-	A way of using the Python interpreter by typing code at the prompt.	-

15.	Script mode	-	A way of using the Python interpreter to read code from a script and run it.	-
16.	Python Interpreter	-	The Python interpreter is a program that reads and executes Python code.	-
17.	Expression	-	A combination of variables, operators, and values that represents a single result.	-
18.	Statement	-	A section of code that represents a command or action.	-
19.	Script	-	A program stored in a file.	-
20.	Linked List	-	A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations.	-
21.	Types of linked list	-	1.Singly linked list 2.Doubly liked list 3.Circular linked list	-
22.	Singly linked list	-	It is a linked list ,in which each node contains only one link field pointing to the next node in the list.	-
23.	Doubly linked list	-	It is a list in which each node has three fields namely data field, forward link and backward link	-
24.	Circularly linked list	-	Circular Linked list is a linked list where all nodes are connected to form a circle. There is no NULL at the end.	-
25.	Applications of a linked list	-	Manipulation of polynomials, sparse matrices, stacks and queues.	-
UNIT –II: STACKS				
26.	Stack	-	Stack is an ordered collection of elements in which insertions and deletions are restricted to one end called top	-
27.	Basic operations that can be performed on a stack	-	• Push operation • Pop operation • Peek operation • Empty check • Fully occupied check	-
28.	Applications of stacks	-	• Towers of Hanoi • Reversing a string • Balanced parenthesis • Recursion using stack • Evaluation of arithmetic expressions	-
29.	Different ways of representing expressions	-	• Infix Notation A+B • Prefix Notation +AB • Postfix Notation AB+	-
30.	Infix notation	-	X + Y: Operators are written in-between their operands. Ex: A * (B + C) / D	-
31.	Prefix Notation	-	Prefix notation is also known as "Polish notation" + X Y: Operators are written before their operands. EX: / * A + B C D	-
32.	Postfix Notation	-	Postfix notation is also known as "Reverse Polish notation" X Y +: Operators are written after their operands. Ex: A B C + * D /	-
33.	Difference between stacks and linked lists	-	The difference between stacks and linked lists is that insertions and deletions may occur anywhere in a linked list, but only at the top of the stack	-
34.	PUSH()	-	push() function is used to insert new elements into	-

			the Stack and pop() function is used to remove an element from the stack	
35.	Pop()	-	Removes an item from the stack. The items are popped in the reversed order in which they are pushed.	-
36.	IsEmpty()	-	Returns true if stack is empty, else false	-
37.	Underflow	-	Underflow occurs when the user performs a pop operation on an empty stack. Overflow occurs when the stack is full and the user performs a push operation. Garbage Collection is used to recover the memory occupied by objects that are no longer used.	-
38.	Overflow	-	Pushing an element into stack already having five elements and stack size of 5, then stack becomes overflow	-
39.	LIFO	-	In stack data structure, elements are added one by one using push operation. Stack follows LIFO Principle i.e. Last In First Out(LIFO).	-
40.	Start pointer	-	In the linked representation of the stack, start pointer behaves as the top pointer variable of stack.	-
41.	Function	-	A named sequence of statements that performs some useful operation. Functions may or may not take arguments and may or may not produce a result.	-
42.	Function Definition	-	A statement that creates a new function, specifying its name, parameters, and the statements it contains.	-
43.	Function object	-	A value created by a function definition. The name of the function is a variable that refers to a function object.	-
44.	Header	-	The first line of a function definition.	-
45.	Body	-	The sequence of statements inside a function definition.	-
46.	Parameter	-	A name used inside a function to refer to the value passed as an argument.	-
47.	Function call	-	A statement that runs a function. It consists of the function name followed by an argument list in parentheses.	-
48.	Argument	-	A value provided to a function when the function is called. This value is assigned to the corresponding parameter in the function.	-
49.	Local variable	-	A variable defined inside a function. A local variable can only be used inside its function.	-
50.	Return value	-	The result of a function. If a function call is used as an expression, the return value is the value of the expression.	-
UNIT-III: QUEUES				
51.	Queue	-	Queue is an ordered collection of elements in which insertions and deletions are restricted to one end.	-
52.	Types of queues	-	• Linear Queues • Circular Queues • Double-Ended-Queue	-
53.	Applications of queues	-	• Jobs submitted to printer • Real life line • Calls to large companies • Access to limited resources in Universities • Accessing files from file server	-
54.	Linear Queues	-	The queue has two ends, the front end and the rear end. The rear end is where we insert elements and front end is where we	-

			delete elements.	
55.	Circular Queue	-	Another form of linear queue in which the last position is connected to the first position of the list.	-
56.	Double-Ended-Queue	-	Another form of queue in which insertions and deletions are made at both the front and rear ends of the queue	-
57.	Priority queue	-	Priority queue is a collection of elements, each containing a key referred as the priority for that element.	-
58.	Applications of priority queue	-	• Operating system • External sorting • Greedy algorithms • Event simulation	-
59.	Over flow	-	Attempt to insert and element when the stack list full is said to be overflow.	-
60.	Under flow	-	Attempt to delete an element when the stack is empty.	-
61.	peek()	-	Gets the element at the front of the queue without removing it.	-
62.	Enqueue	-	It is used add a new element into a queue. Enqueue operation is performed at the end of the list. It allocates memory for the new node.	-
63.	Output Restricted Queue	-	A queue which satisfies the following properties • Insertion is allowed at both ends • Deletion takes place at one end	-
64.	Input Restricted Queue	-	A queue which satisfies the following properties • Insertion takes place at one end • Deletion is allowed at both ends	-
65.	NumPy	-	Numerical Python is used for efficient and general numeric computations on numerical data saved in arrays. E.g., sorting, indexing, reshaping, and more.	-
66.	SciPy	-	Scientific Python is a collection of tools in Python used to perform operations such as integration, differentiation, and more.	-
67.	Tkinter	-	It is an in-built Python module that is used to create GUI application.	-
68.	Object-Oriented Programming	-	A style of programming in which data and the operations that manipulate it are organized into classes and methods.	-
69.	Object	-	An object in Python is defined as an instance that has both state and behaviour. Everything in Python is made of objects.	-
70.	Class	-	Class is defined as a logical entity that is a huge collection of objects and it also contains both methods and attributes.	-
71.	Attribute	-	One of the named values associated with an object.	-
72.	Polymorphism	-	Polymorphism means the ability to take multiple forms.	-
73.	Encapsulation	-	Encapsulation means binding the code and the data together.	-
74.	Data Abstraction	-	Data Abstraction is providing only the required details and hiding the implementation from the world.	-

75.	Pandas	-	Pandas is a Python library that provides highly flexible and powerful tools and high-level data structures for analysis.	-
UNIT-IV: TREES				
76.	Tree	-	A tree is a non-linear data structure, which represents hierarchical relationship between individual data items	-
77.	Path	-	A path in a tree is a sequence of distinct nodes in which successive nodes are connected by edges in the tree	-
78.	Terminal nodes	-	A node that has no children is called as a terminal node. It is also referred as a leaf node. These nodes have degree has zero	-
79.	Binary tree	-	A binary tree is a tree in which every non-leaf node has at most two children	-
80.	Full binary tree	-	A full binary tree is a tree in which all leaves are on the same level and every non-leaf node has exactly two children	-
81.	Complete binary tree	-	A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible.	-
82.	Tree traversal	-	Traversing a binary tree, means moving through all the nodes in the binary tree, visiting each node in the tree only once	-
83.	Min-heap	-	Every node n except the root, has a value greater than its parent is referred to as min heap	-
84.	Max-heap	-	Every node n except the root, has a value lesser than its parent is referred to as max heap	-
85.	Root	-	Node in the tree in which further sub trees were attached. A root node of a tree has its child.	-
86.	Degree of Node	-	The total number of sub tree attached to that node is called the degree of the node.	-
87.	Leaf Nodes	-	These are the terminals nodes of the tree. The nodes which have degree 0 are called as leaf nodes of the tree	-
88.	Internal Nodes	-	The nodes in the tree which are other than leaf nodes and the root node are called as internal nodes.	-
89.	Parent Node	-	The node which is having further sub branches is called the parent node of those sub branches	-
90.	Predecessor	-	if some particular nodes previous to some other nodes than that node is called the predecessor of the other node	-
91.	Level of tree	-	The root node is always considering at level zero, and then its adjacent children are supposed to be at level 1 and so on	-
92.	Height of the tree	-	The maximum level is the height of the tree	-
93.	Forest	-	A tree may be defined as a forest in which only a single node (root) has no predecessor Any forest is consist of collection of trees.	-
94.	Degree of tree	-	The maximum degree of the node in the tree is called the degree of the tree.	-
95.	Traversal	-	process of visiting (checking and/or updating) each node in a tree data structure, exactly once	-
96.	Inorder Traversal	-	Visit Left-Root-Right	-
97.	Preorder Traversal	-	Visit Root-Left-Right	-
98.	Postorder Traversal	-	Visit Left-Right-Root	-

99.	Siblings	-	If they are present at the same level, and their parents are same.	-
100.	Search	-	Searches an element in a tree.	-
UNIT-V: SEARCHING AND SORTING				
101.	Hashing	-	Hashing is the transformation of string of characters into a usually shorter fixed length value or key that represents the original string.	-
102.	Hash table	-	The hash table data structure is an array of some fixed size, containing the keys. A key is a string with an associated value.	-
103.	Hash function	-	A simple hash function is **hash_key=key mod tablesize**.	-
104.	Extendible hashing	-	A hash table in which the hash function is the last few bits of the key and the table refer to buckets.	-
105.	Collision	-	When an element is inserted, it hashes to the same value as an already inserted element, and then it produces collision.	-
106.	Collision resolution Techniques	-	• Separate chaining • Open Addressing • Rehashing	-
107.	Separate chaining	-	Separate chaining is a collision resolution technique to keep the list of all elements that hash to the same value.	-
108.	Advantages of separate chaining	-	More number of elements can be inserted as it uses linked lists	-
109.	Open addressing	-	Open addressing is a collision resolving strategy in which, if collision occurs alternative cells are tried until an empty cell is found.	-
110.	Probing	-	Probing is the process of getting next available hash table array cell	-
111.	Types of collision resolution	-	• Linear probing • Quadratic probing	-
112.	Limitations of linear probing	-	• Time taken for finding the next available cell is large. • In linear probing, we come across a problem known as clustering.	-
113.	Searching	-	Finding an element position in a given array called searching	-
114.	Types of searching	-	• Linear search • Binary search	-
115.	Linear Search	-	Sequentially moves through your collection (or data structure) looking for a matching value	-
116.	Binary search	-	Binary search, also known as a half-interval search.	-
117.	Sorting	-	Sorting arranges data in a sequence which makes searching easier	-
118.	Types of Sorting	-	1. Bubble Sort 2. Insertion Sort 3. Selection Sort 4. Quick Sort 5. Merge Sort 6. Heap Sort	-
119.	Bubble Sort	-	Sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order.	-
120.	Insertion Sort	-	Insertion sort is a sorting algorithm in which the elements are transferred one at a time to the right position.	-

121.	Selection Sort	-	Place comparison-based algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end.	-
122.	Quick Sort	-	It picks an element as pivot and partitions the given array around the picked pivot.	-
123.	Merge Sort	-	It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves.	-
124.	Heap Sort	-	It is similar to selection sort where we first find the maximum element and place the maximum element at the end.	-
125.	Radix sort	-	It is the generalized term of bucket sort it can be performed using buckets from 0 to 9.	-
Placement Questions				
126.	Difference between file structure and storage structure	-	**Storage structure:** It is the representation of the data structure in the computer memory. **File structure:** It is the representation of the storage structure in the auxiliary memory.	-
127.	Applications of Stack data structure	-	• Expression evaluation • Backtracking • Memory Management • Function calling and return	-
128.	Stack overflow condition.	-	Overflow occurs when **top = Maxsize -1**	-
129.	One-dimension array	-	Each value is called an element of the array. The elements of the array share the same variable name but each element has its own unique index number	-
130.	Heterogeneous linked list	-	A linked list that can hold nodes of different types at the same time.	-
131.	Scenarios Circular queue	-	• If (rear + 1)%maxsize = front, the queue is full. In that case, overflow occurs and therefore, insertion cannot be performed in the queue. • If rear!= max - 1, the rear will be incremented to the mod(maxsize) and the new value will be inserted at the rear end of the queue. • If front != 0 and rear = max - 1, it means that queue is not full therefore, set the value of rear to 0 and insert the new element there.	-
132.	Minimum number of queues needed to implement the priority queue	-	Two queues are needed. One queue is used to store the data elements, and another is used for storing priorities.	-
133.	Maximum number of nodes in a binary tree of height k	-	2^{k+1}-1 where k >= 1	-
134.	Data structure suits the tree construction	-	Queue data structure	-

135.	Data structures are used in BFS and DFS algorithms	-	In BFS algorithm, Queue data structure is used. In DFS algorithm, Stack data structure is used.	-
136.	Scenario of Binary Search	-	Binary Search algorithm is used to search an already sorted list. The algorithm follows divide and conquer approach	-
137.	Applications of Multilinked Structures	-	• Sparse matrix, • Index generation.	-
138.	Difference between NULL and VOID	-	• Null is actually a value, whereas Void is a data type identifier. • A null variable simply indicates an empty value, whereas void is used to identify pointers as having no initial size.	-
139.	Implementation Doubly linked list	-	Doubly linked list can be implemented using a single pointer.	-
140.	Implement a stack using queue	-	Stack 's' can be implemented in two ways: • Method 1 (By making push operation costly) • Method 2 (By making pop operation costly)	-
141.	Applications of A Circular Queue	-	• Memory management • Process Scheduling • Traffic Systems	-
142.	Dynamic memory allocation helps in managing data	-	Apart from being able to store simple structured data types, dynamic memory allocation can combine separately allocated structured blocks to form composite structures that expand and contract as needed.	-
143.	Minimum number of nodes in binary tree	-	A binary tree can have a minimum of zero nodes, which occurs when the nodes have NULL values. Furthermore, a binary tree can also have 1 or 2 nodes	-
144.	Declaration statements result in a fixed reservation	-	Most declarations do, with the exemption of pointers. Pointer declaration does not allocate memory for data, but for the - address of the pointer variable.	-
145.	Search a target key in a linked list	-	To find the target key in a linked list, have to apply sequential search.	-
146.	Process of Selection sort	-	Selection sort works by picking the smallest number from the list and placing it at the front. This process is repeated for the second position towards the end of the list. It is the simplest sort algorithm.	-
147.	Minimum number of nodes in binary tree	-	A binary tree can have a minimum of zero nodes, which occurs when the nodes have NULL values. Furthermore, a binary tree can also have 1 or 2 nodes	-
148.	Ordered list	-	An ordered list is a list in which each node's position in the list is determined by the value of its key component, so that the key values form an increasing sequence, as the list is traversed.	-
149.	Application of tree data-structure	-	• The manipulation of Arithmetic expression, • Symbol Table construction, • Syntax analysis.	-
150.	Data structure in recursion	-	Operating system maintains the stack in order to save the iteration variables at each function call	-

	Faculty Team Prepared	**Signature**
1.	**Mr.S.Gopi, AP/IT**	
2.	**Mr.T.Manivel, AP/IT**	
3.	**Mrs.A.Kalaiyarasi, AP/IT**	

HoD

Estd. 2000