TUTORING AND TESTING CENTER
NOVA SOUTHEASTERN UNIVERSITY

Academic Resource Sheet

CSIS 3400 Resources - Data Structures

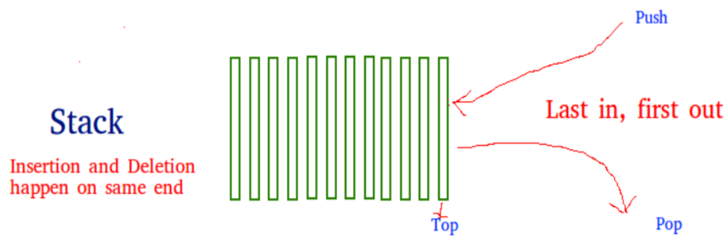
Pointers in C++:

*A pointer is a variable that stores the memory address as its value. It points to a data type (like int or string) of the same type and is created with the * operator. The address of the variable you're working with is assigned to the pointer. Example:*

```
string food = "Pizza";  // A food variable of type string
string* ptr = &food;    // A pointer variable, with the name ptr, that stores the address of food

// Output the value of food (Pizza)
cout << food << "\n";

// Output the memory address of food (0x6dfed4)
cout << &food << "\n";

// Output the memory address of food with the pointer (0x6dfed4)
cout << ptr << "\n";
```

Stack Data Structure:

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO (Last In First Out) or FILO (First In Last Out).
There are many real-life examples of a stack. Consider an example of plates stacked over one another in the canteen. The plate which is at the top is the first one to be removed, i.e. the plate which has been placed at the bottommost position remains in the stack for the longest period of time.

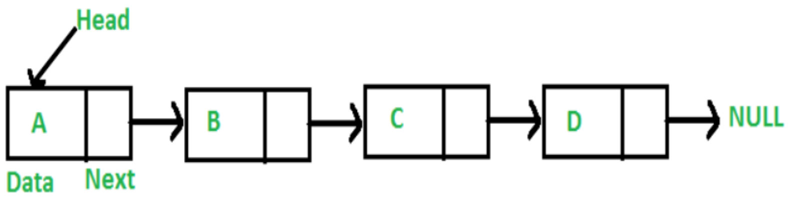


Stack VS Linked List:

STACK	LINKED LIST
An abstract data type that serves as a collection of elements with two principal operations which are push and pop	A linear collection of data elements whose order is not given by their location in memory
Push, pop and peek are the main operations performed on a stack	Insert, delete and traversing are the main operations performed on a linked list
Can read the topmost element	It is required to traverse through each element to access a specific element
Works according to the FIFO mechanism	Elements connect to each other by references
Simpler than linked list	More complex than stack

Linked List Data Structure:

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown below:

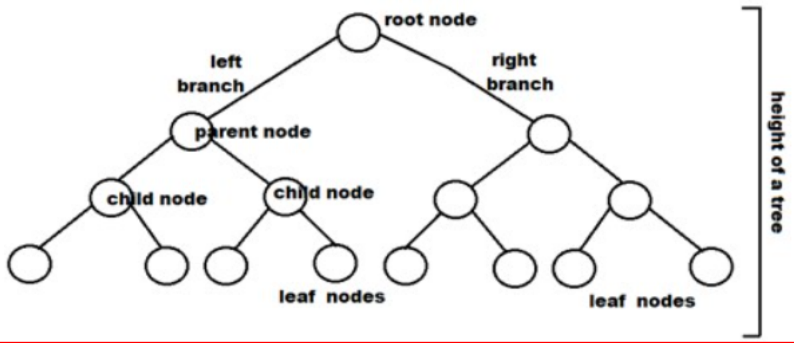


Linked List VS Arrays:

ARRAY	LINKED LISTS
1. Arrays are stored in contiguous location.	1. Linked lists are not stored in contiguous location.
2. Fixed in size.	2. Dynamic in size.
3. Memory is allocated at compile time.	3. Memory is allocated at run time.
4. Uses less memory than linked lists.	4. Uses more memory because it stores both data and the address of next node.
5. Elements can be accessed easily.	5. Element accessing requires the traversal of whole linked list.
6. Insertion and deletion operation takes time.	6. Insertion and deletion operation is faster.

Binary Tree Data Structure:

A binary tree is a tree-type non-linear data structure with a maximum of two children for each parent. Every node in a binary tree has a left and right pointer along with the data element. The node at the top of the hierarchy of a tree is called the root node. The nodes that hold other sub-nodes are the parent nodes. A parent node has two child nodes: the left child and right child.



Contact us via:

Student Affairs Building, 2nd floor
(954) 262-8350
@nsu_ttc @nsu_si