



Data Structures Class 12 Notes | Computer Science

Data Structures Class 12 Notes

Introduction to Data Structure

A data structure is a way how the data is organised and stored in a computer. A data structure is like a container that holds a group of data which can be processed as a single unit. Data structures make it simple to organise, search and work with data quickly and efficiently.

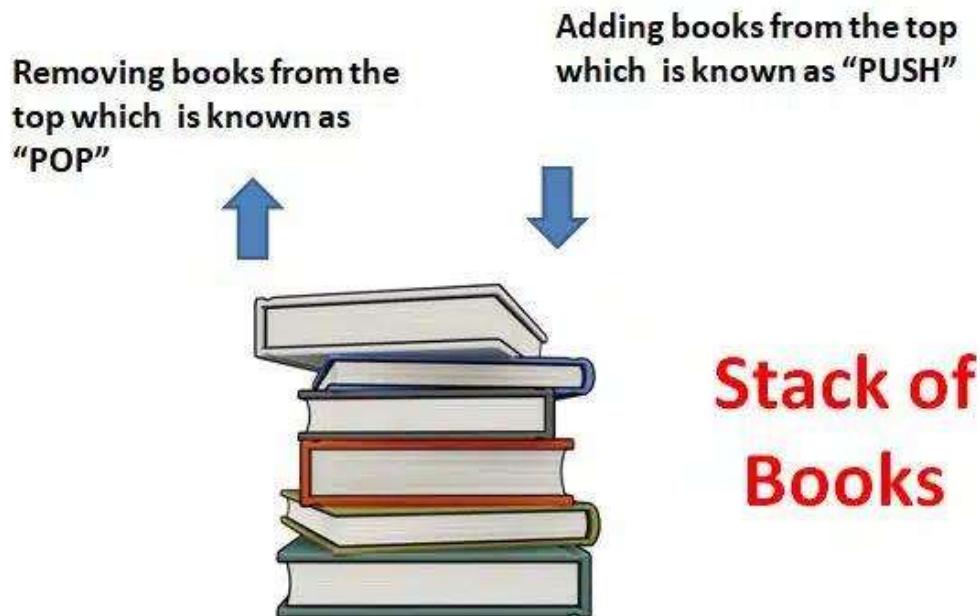
Type of data structure

There are two different types of data structure:

- **Linear data structure:** In a linear data structure, data is arranged in a straight line, one after the other. Example: arrays, stacks and queues.
- **Non-linear data structure:** In a non-linear data structure, data is organised in a hierarchical method, like a tree. Example: tree and graphs.

What is stack?

A stack is a linear data structure which allows adding and removing elements in a Last In, First Out order. This means that the recently added elements will be removed first. For example, with multiple books on the table in stack position, you can add or remove a books from the top, which is known as Last In, First Out.

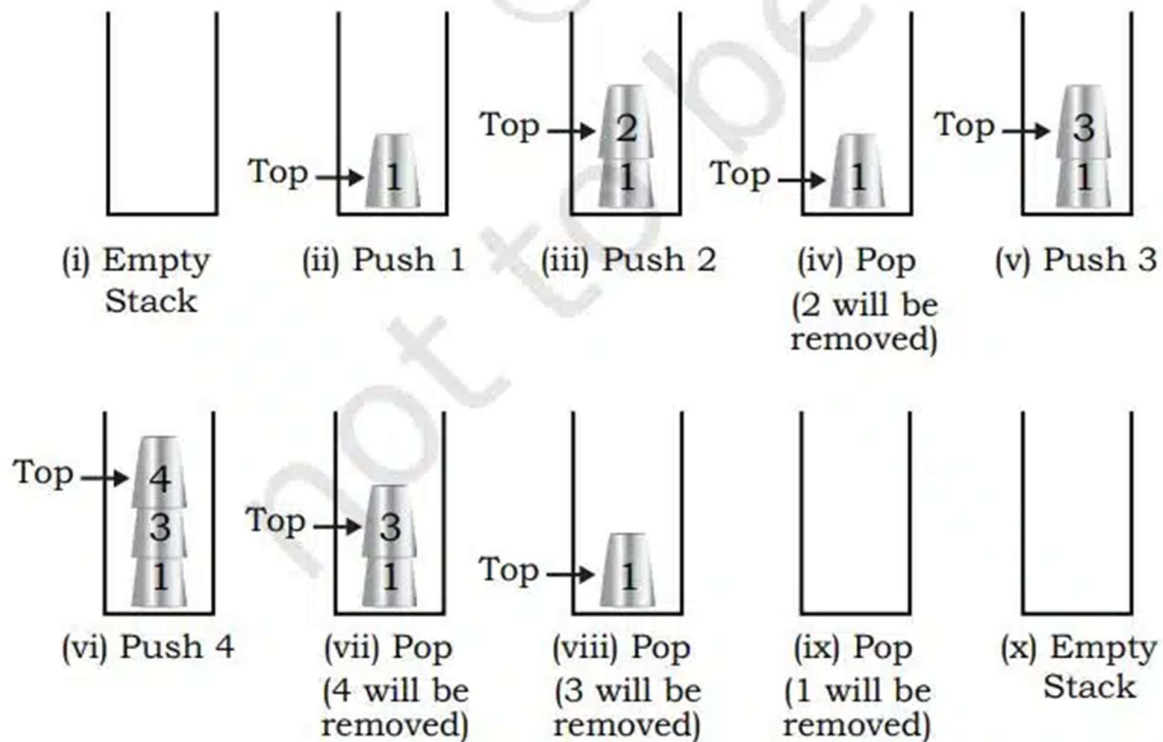


Why we use a Stack

- Stack helps to manage the data in a Last In, First Out method.
- It helps to handle memory allocation.
- Managing browser history
- Compilers and interpreters use the stack method for converting them into machine language.

Operations on stack (push & pop)

PUSH adds a new element at the TOP of the stack. A stack is full when no more elements can be added to it and is known as 'overflow'. POP operation is used to remove the topmost element of the stack. Trying to delete an element from an empty stack is known as 'underflow'. A stack is used to insert and delete elements in LIFO order.



implementation of stack using list

Let us write a program to create a STACK:

- Insert/delete elements
- Check if the STACK is empty
- Find the number of elements in the STACK
- Read the value of the topmost element in the STACK

```
# Initialize an empty
stack glassStack = list()

# Function to push an element onto the
stack def opPush(stack, element):
    stack.append(element)
    print(f"Pushing element: {element}")

# Function to pop an element from the
stack def opPop(stack):
    if stack:
        return
    stack.pop() else:
        return None

# Function to get the size of the stack
def size(stack):
    return len(stack)

# Function to get the top element of the
stack def top(stack):
```

```

    if stack:
        return stack[-1]
    else:
        return None

# Function to display all elements in the
stack def display(stack):
    print("Stack elements:", stack)

# Add elements to the stack
element = 'glass1'
opPush(glassStack, element)
element = 'glass2'
opPush(glassStack, element)

# Display number of elements in the stack
print("Current number of elements in stack:", size(glassStack))

# Delete an element from the stack
element = opPop(glassStack)
if element:
    print("Popped element:", element)

# Add a new element to the
stack element = 'glass3'
opPush(glassStack, element)

# Display the last element added to the stack
print("Top element:", top(glassStack))

# Display all elements in the
stack display(glassStack)

# Delete all elements from the stack
while True:
    item = opPop(glassStack)
    if item is None:
        print("Stack is empty now.")
        break
    else:
        print("Popped element:", item)

```

Output:

```

Pushing element: glass1
Pushing element: glass2
Current number of elements in stack: 2
Popped element: glass2
Pushing element: glass3
Top element: glass3
Stack elements: ['glass1', 'glass3']
Popped element: glass3
Popped element: glass1
Stack is empty now.

```