FULL STACK

Data Structures: Arrays and Lists

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
   cout << "Hello, world!" << endl;
   return 0;
}
```

©Simplilearn. All rights reserved.

simplilearn

You Already Know

Course(s):

1. Beginning Java Data Structures and Algorithms

2. Maven

©Simplilearn. All rights reserved.

Recap

- Explain the fundamentals of data structures
  - What is data structure?
  - Fundamentals of data structure

- Explain the fundamentals of Maven
  - Maven build life cycle

- Set up your first Maven project
  - Building a jar

©Simplilearn. All rights reserved.

Recap

- Build a jar
  - Building a jar

- Demonstrate Maven plugins
  - Play with Maven plugins

- Download and include the Maven dependencies in the project
  - Maven POM
  - Scope of a dependency
  - Versioning of dependencies

©Simplilearn. All rights reserved.

A Day in the Life of a Full Stack Developer

In the last sprint, Joe has done an excellent job. He has developed a program that stores all the errors, warnings, and exceptions in an error log file that are raised when the users work with the website.

After noticing his remarkable performance, the organization has decided to hand over the most important project to Joe.

He has been asked to develop a program where consecutive log files can be accessed from the cache rather the actual database.

In this lesson, we will learn how to solve this real-world scenario to help Joe complete his task effectively and quickly.

©Simplilearn. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- Explain arrays
- Implement array rotations, order statistics, range queries, and matrix
- Explain linked lists
- Demonstrate singly linked list, circular linked list, and doubly linked list
- Design and implement the stack
- Design and implement the queue

©Simplilearn. All rights reserved.

FULL STACK

Data Structures

©Simplilearn. All rights reserved.

Data Structures

Data structures are specialized ways to store and retrieve the data.

Data structures are used to gather information from the user to process, maintain, and retrieve the information.

General data structure types are arrays, stacks, queues, and trees.

Appropriate data structures help the code run smoothly.

©Simplilearn. All rights reserved.

Arrays

©Simplilearn. All rights reserved.

Arrays

An array is a collection of elements of the same data type stored at contiguous memory locations. It is index-based. The first element refers to index 0.

Single-dimensional Arrays

In single-dimensional arrays, elements are stored in rows only.

Example:

int a= new int[5];

a[0]	a[1]	a[2]	a[3]	a[4]

Multidimensional Arrays

In multidimensional arrays, elements are stored in the form of rows and columns.

Example:

int s [ ] [ ]=new int [3] [3]

S[0][1]	S[0][1]	S[0][2]
S[1][0]	S[1][1]	S[1][2]
S[2][0]	S[2][0]	S[2][2]

©Simplilearn. All rights reserved.

Array Rotation

Duration: 20 min.

Problem Statement:

Write a program to demonstrate array rotation.

ASSISTED PRACTICE

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines

Steps to demonstrate array rotation:

1. Create a Java project in your IDE
2. Write a program in Java to demonstrate array rotation
3. Initialize the .git file
4. Add and commit the program files
5. Push the code to your GitHub repositories

©Simplilearn. All rights reserved.

Order Statistics

Duration: 20 min.

ASSISTED PRACTICE

Problem Statement:

Write a program to demonstrate the order statistics.

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines

Steps to demonstrate order statistics:

1. Create a Java project in your IDE
2. Write a program in Java to demonstrate order statistics
3. Initialize the .git file
4. Add and commit the program files
5. Push the code to your GitHub repositories

©Simplilearn. All rights reserved.

simplilearn

Range Queries

Duration: 20 min.

Problem Statement:

Write a program to demonstrate range queries.

ASSISTED PRACTICE

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines

Steps to demonstrate range queries:

1. Create a Java project in your IDE
2. Write a program in Java to demonstrate range queries
3. Initialize the .git file
4. Add and commit the program files
5. Push the code to your GitHub repositories

©Simplilearn. All rights reserved.

Matrix

Duration: 20 min.

Problem Statement:

Write a program to demonstrate working of matrices.

ASSISTED PRACTICE

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines

Steps to demonstrate matrix:

1. Create a Java project in your IDE
2. Write a program in Java to demonstrate working of matrices
3. Initialize the .git file
4. Add and commit the program files
5. Push the code to your GitHub repositories

©Simplilearn. All rights reserved.

FULL STACK

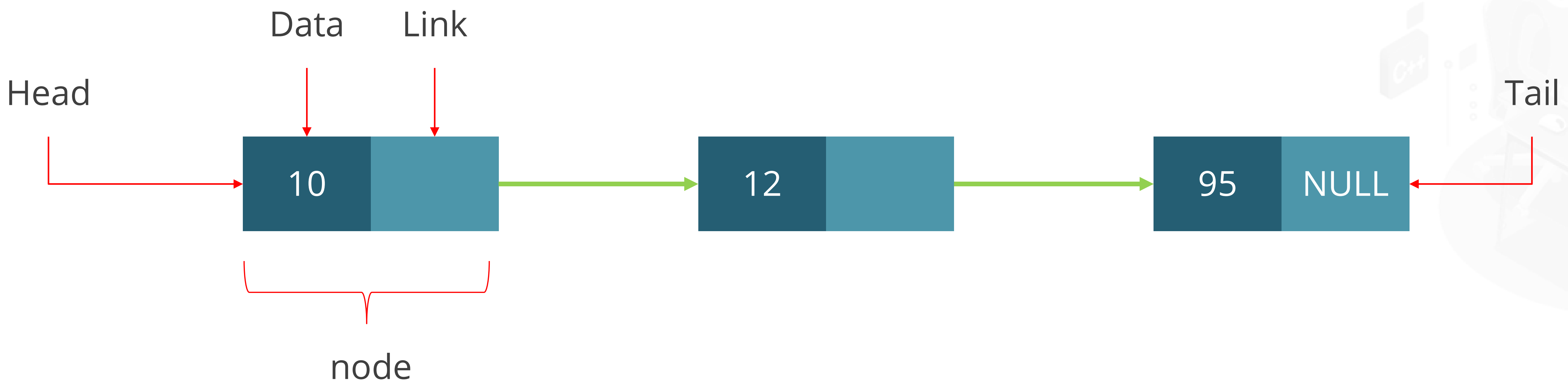
Linked List

©Simplilearn. All rights reserved.

Linked List

- The linked list is a collection of nodes that are not stored at contiguous memory locations.

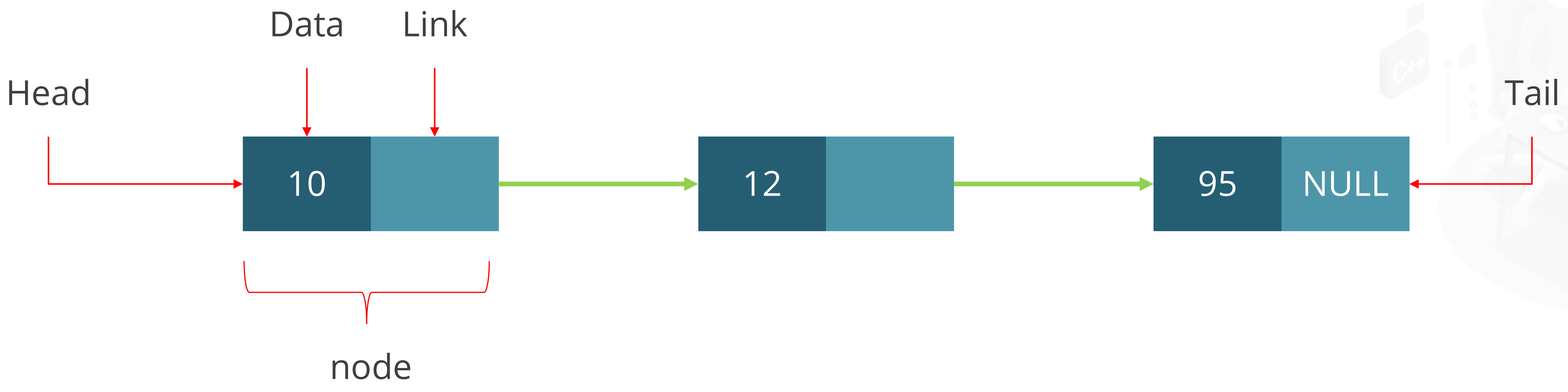
- Every node contains two fields. One field is used to store the data and the other field contains the address of the next node in the memory.



©Simplilearn. All rights reserved.

Linked List

- The last node of the linked list contains the pointer to NULL.
- The empty nodes cannot be present in the linked list.
- The memory size of the linked list is limited and does not need to be declared in advance.



©Simplilearn. All rights reserved.

FULL STACK

Singly Linked List

©Simplilearn. All rights reserved.

Singly Linked List

Every node contains two fields: data field and pointer to the next node.

It is the most commonly used linked list.

It is the collection of an ordered set of elements.

It can move only in one direction.

©Simplilearn. All rights reserved.

Singly Linked List

Duration: 20 min.

Problem Statement:

Write a program to create and perform operations on the singly linked list.

ASSISTED PRACTICE

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines

Steps to demonstrate working of singly linked list:

1. Create a Java project in your IDE
2. Write a program in Java to create a singly linked list
3. Initialize the .git file
4. Add and commit the program files
5. Push the code to your GitHub repositories

©Simplilearn. All rights reserved.

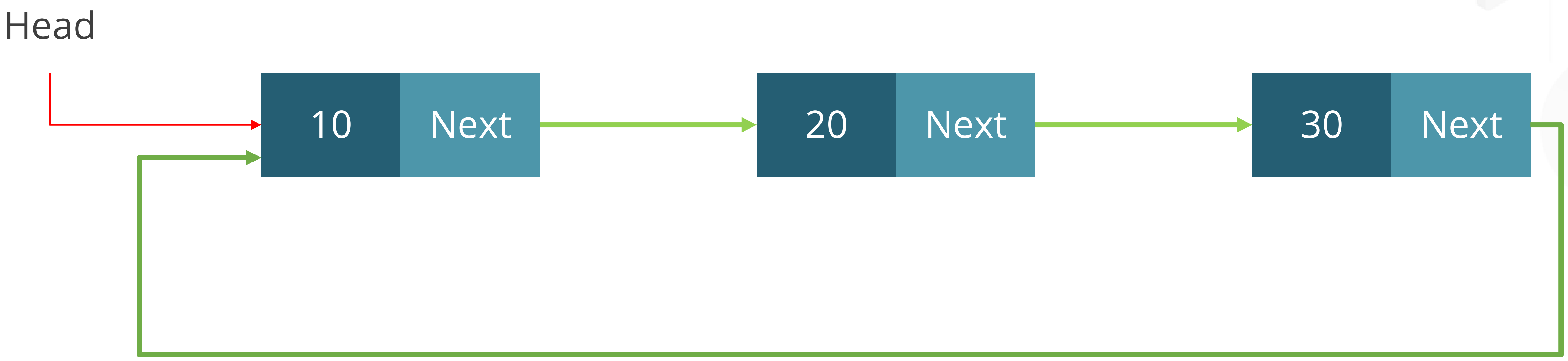
FULL STACK

Circular Linked List

©Simplilearn. All rights reserved.

Circular Linked List

- In this type of a linked list, the last node of the list has the pointer to the first node.
- It does not have a beginning or an end.
- Circular links go around the nodes until the program reaches the same node from where it started.



©Simplilearn. All rights reserved.

Circular Linked List

Duration: 20 min.

Problem Statement:

Write a program to create and perform operations on the circular linked list.

ASSISTED PRACTICE

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines

Steps to demonstrate the working of circular linked list:

1. Create a Java project in your IDE
2. Write a program in Java to create a circular linked list
3. Initialize the .git file
4. Add and commit the program files
5. Push the code to your GitHub repositories

©Simplilearn. All rights reserved.

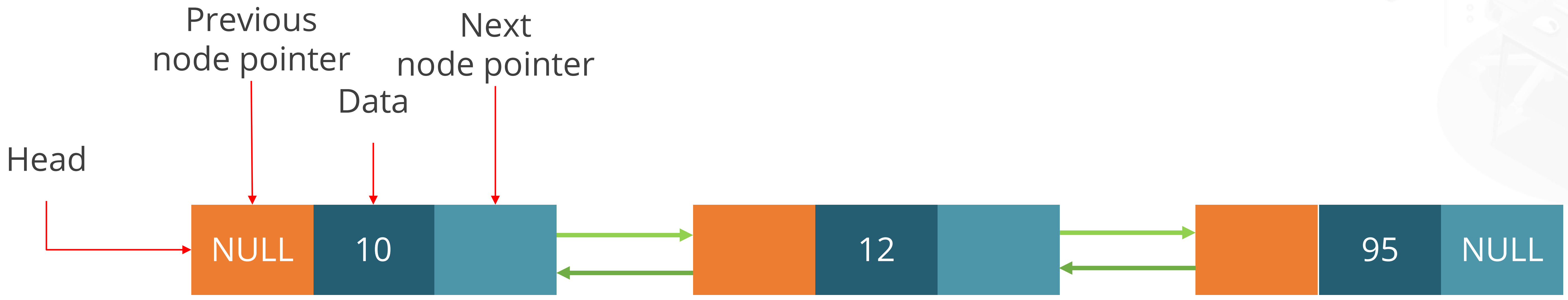
Doubly Linked List

©Simplilearn. All rights reserved.

Doubly Linked List

- In a doubly linked list, each node contains the pointer to the previous as well as to the next node in the sequence.

- In this type of a linked list, the pointer can move in both the directions.



©Simplilearn. All rights reserved.

Doubly Linked List

Duration: 20 min.

Problem Statement:

Write a program to create and perform operations on the doubly linked list.

ASSISTED PRACTICE

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines

Steps to demonstrate the working on doubly linked list:

1. Create a Java project in your IDE
2. Write a program in Java to create a doubly linked list
3. Initialize the .git file
4. Add and commit the program files
5. Push the code to your GitHub repositories

©Simplilearn. All rights reserved.

FULL STACK

Stack

©Simplilearn. All rights reserved.

Stack

Stack

Stack is a linear data structure in which insertion and deletion of elements can be done only at one end called the top.

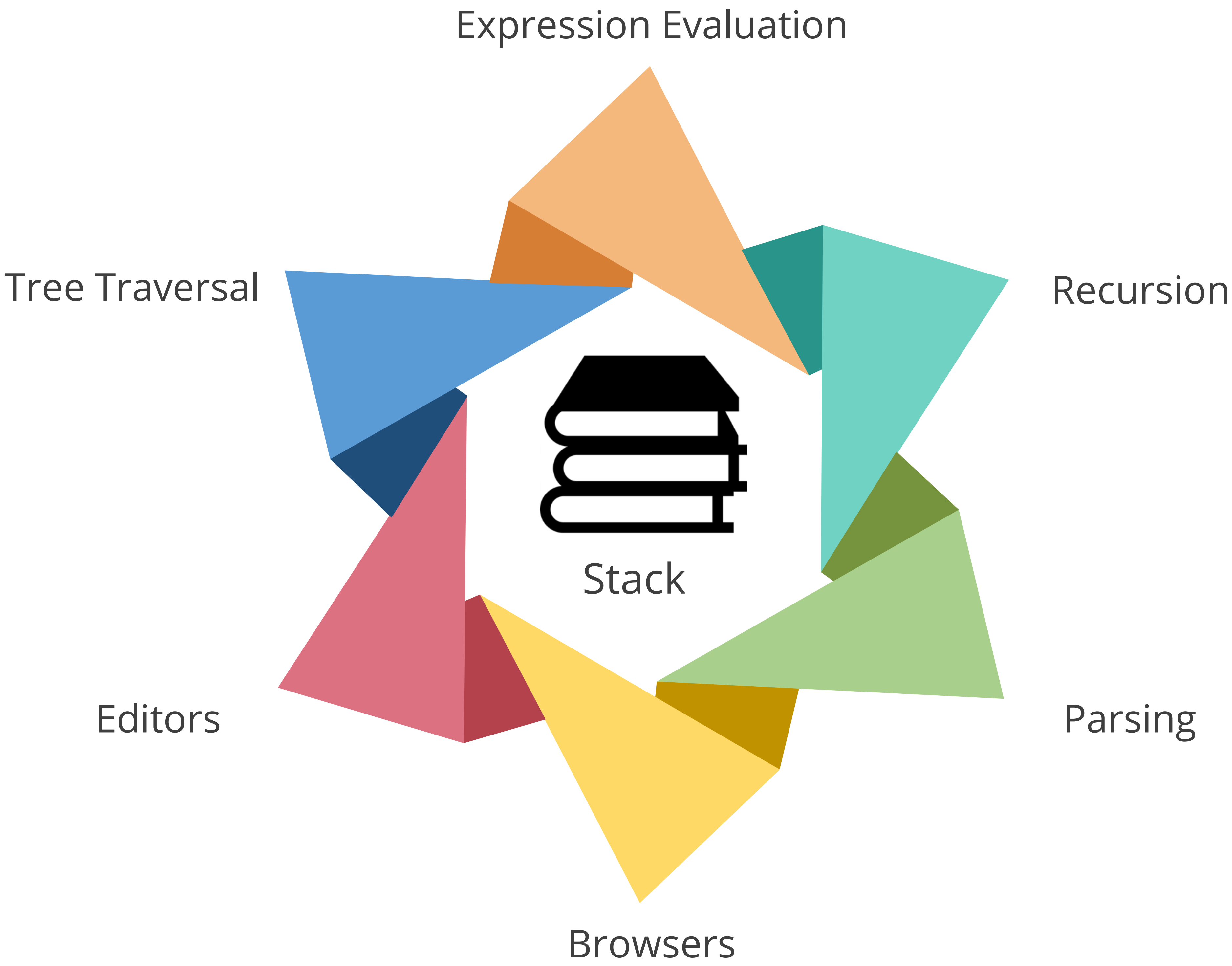
It has a pointer to its top element.

It works in a First In Last Out (FILO) manner.

Example: Coins stacked over one another

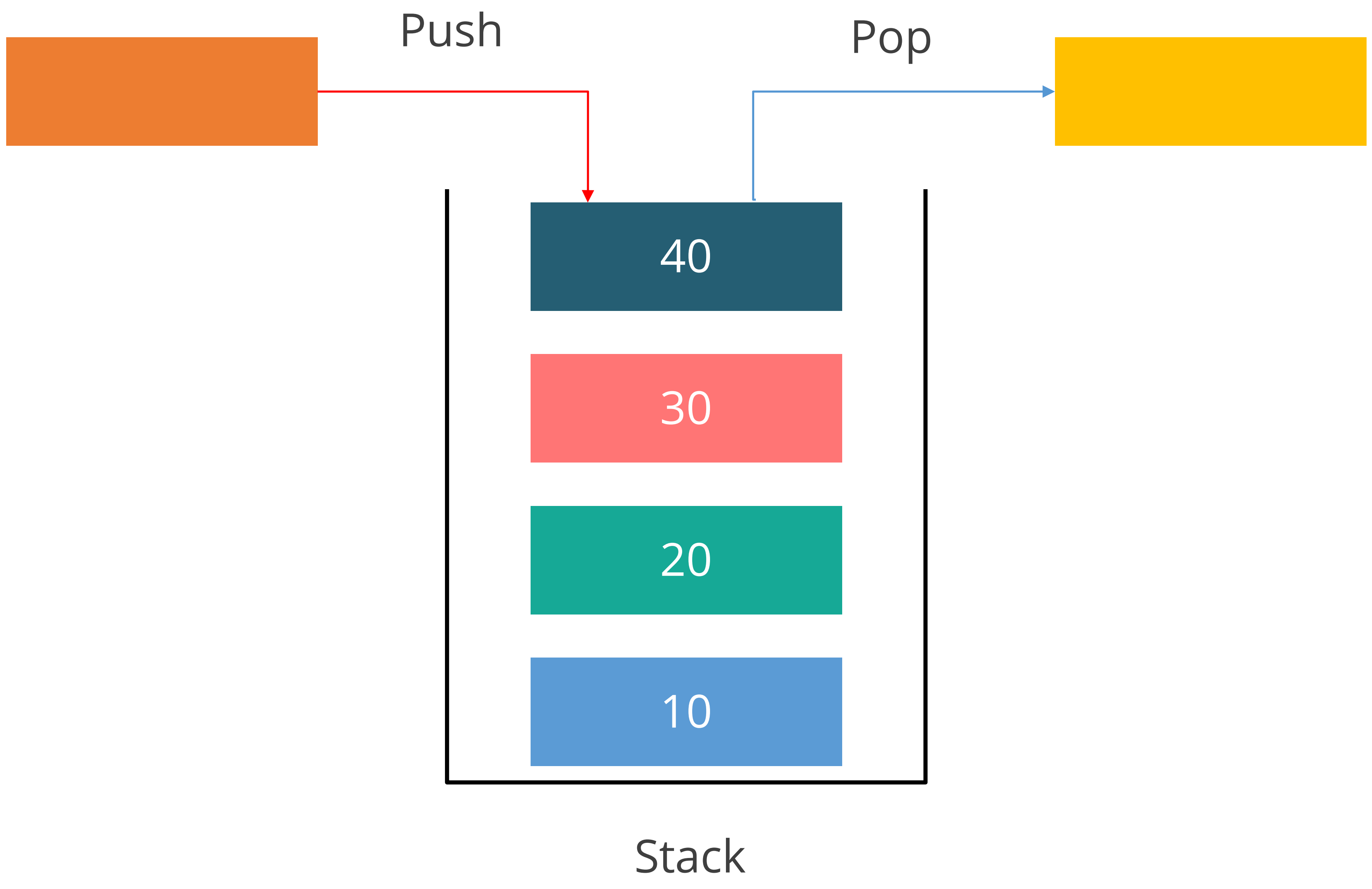
©Simplilearn. All rights reserved.

Applications of the Stack



©Simplilearn. All rights reserved.

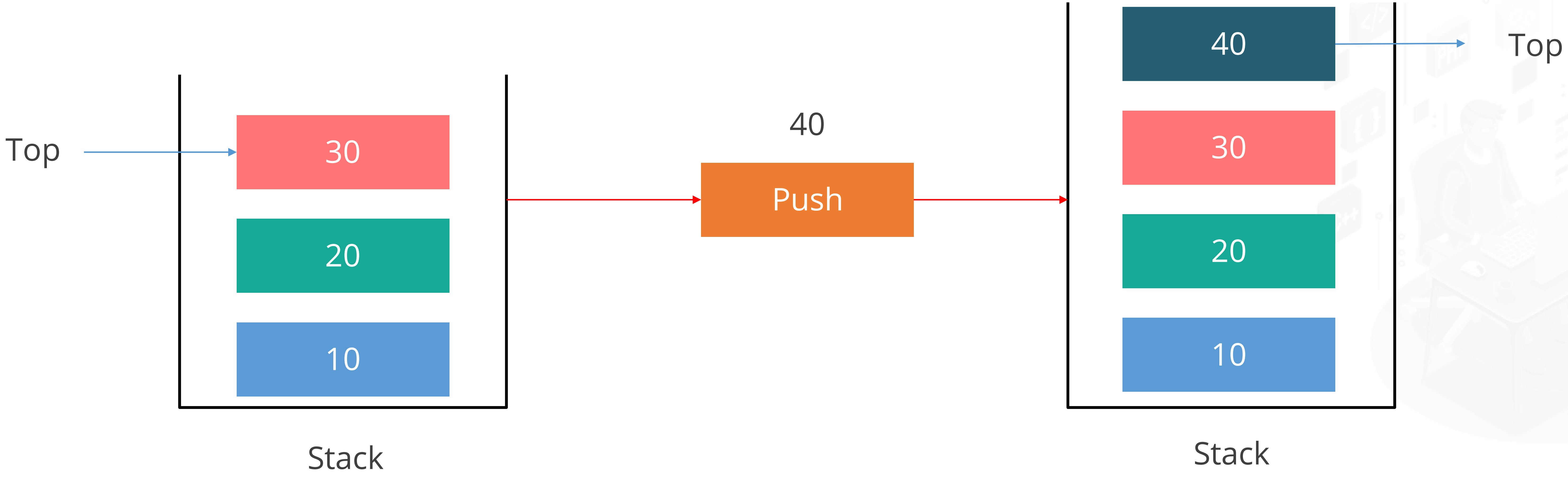
Operations Performed on a Stack



©Simplilearn. All rights reserved.

Operations Performed on a Stack

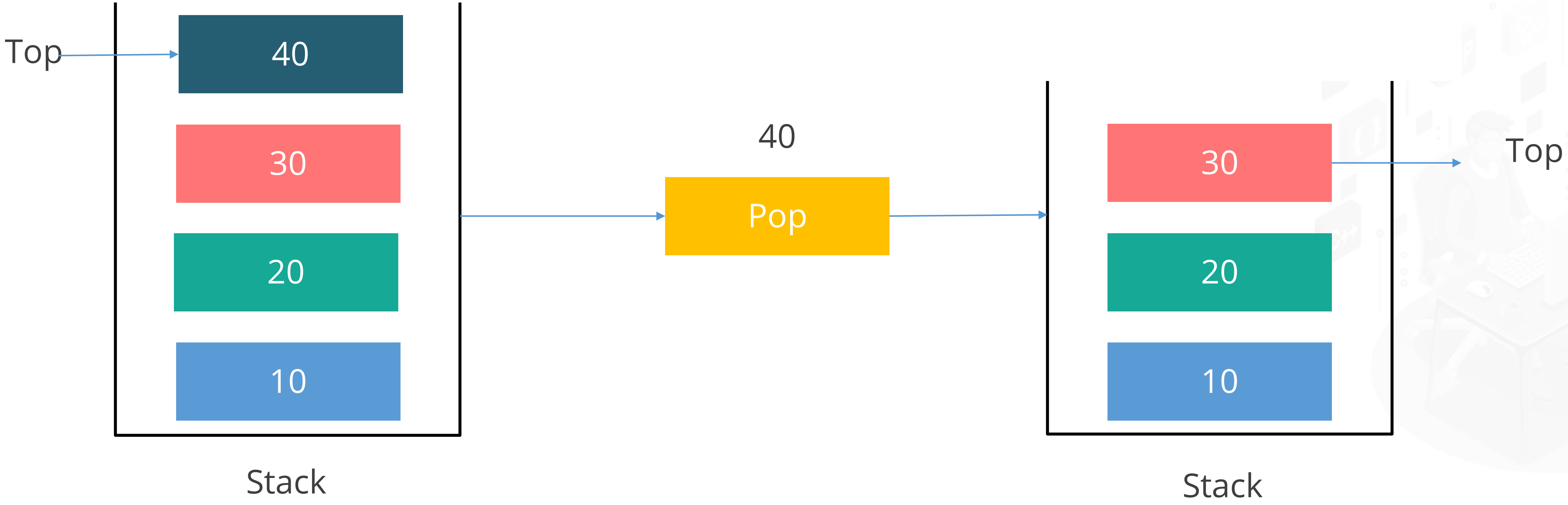
Adding an element to the stack:



©Simplilearn. All rights reserved.

Operations Performed on a Stack

Removing an element from the stack:



©Simplilearn. All rights reserved.

Top and Its Values

While working on the stack, you have to check the value of the top. According to that, you will get to know the status of the stack.

Top Value	Stack Status
-1	The stack is empty.
0	There is only one element in the stack.
N-1	The stack is full.
N	The stack gets overflowed.

©Simplilearn. All rights reserved.

Stack

Duration: 20 min.

Problem Statement:

Write a program to perform operations on a stack.

ASSISTED PRACTICE

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines

Steps to demonstrate functionality of stack:

1. Create a Java project in your IDE
2. Write a program in Java to create a stack and perform operations on it
3. Initialize the .git file
4. Add and commit the program files
5. Push the code to your GitHub repositories

©Simplilearn. All rights reserved.

FULL STACK

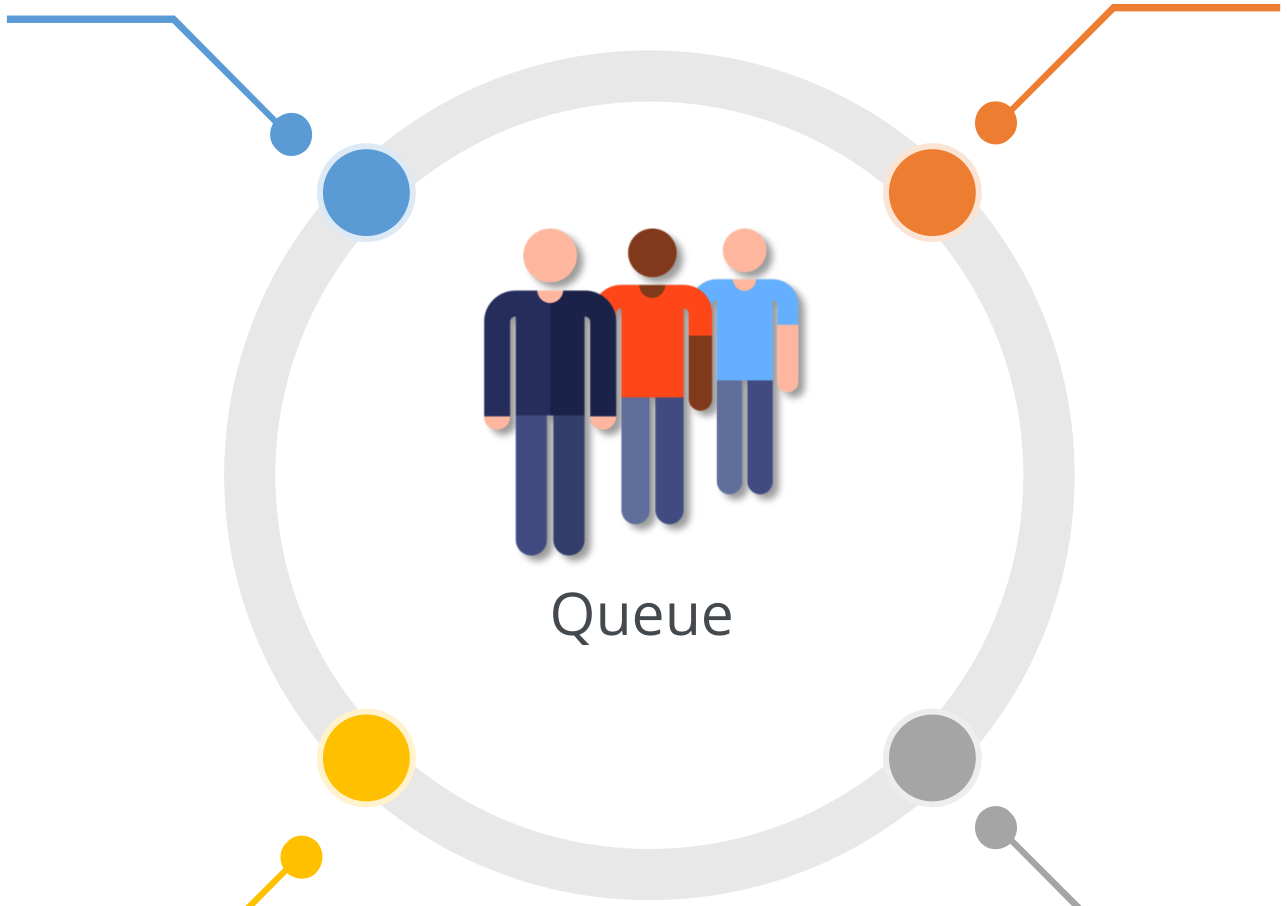
Queue

©Simplilearn. All rights reserved.

Queue

A queue is a linear structure which enables the user to insert the elements from the ***REAR*** end and delete from the ***FRONT*** end.

It works in a First In First Out (FIFO) manner.



The difference between the stack and the queue is the way elements are removed.

Example: People waiting in a queue to vote.

©Simplilearn. All rights reserved.

Applications of the Queue

To perform asynchronous data transfer

To handle interrupts in operating systems

To add or remove the songs in a playlist

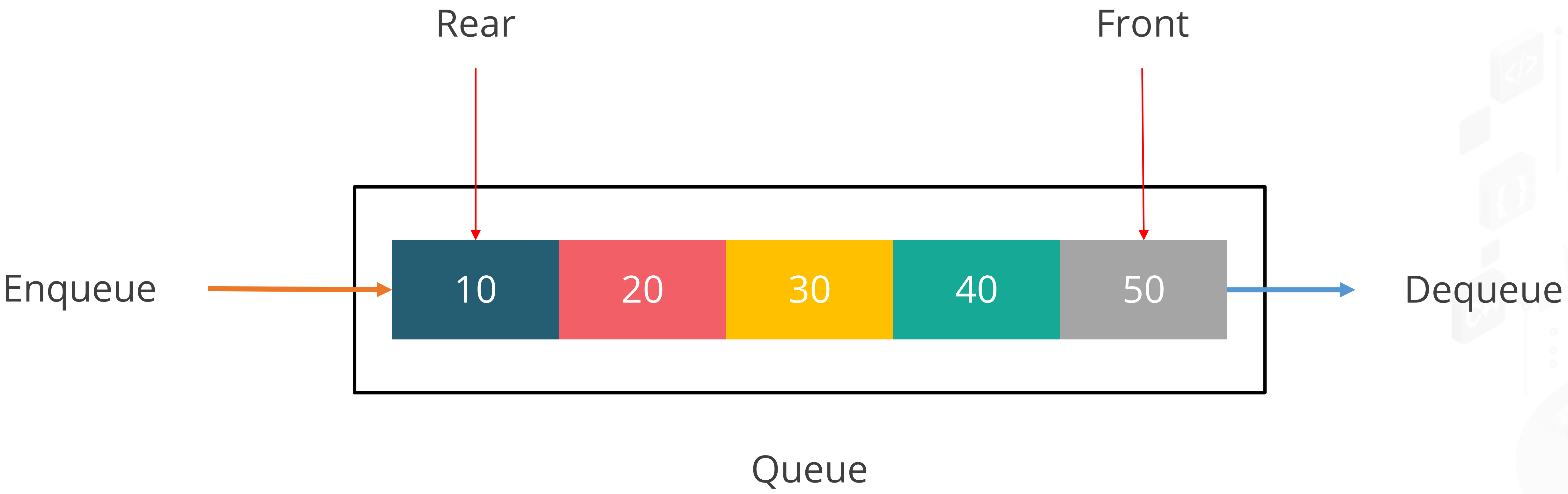
To arrange the data in a sequence in an MP3 or a CD player

To assign waiting lists for single-shred resources like a printer or a disk

To achieve synchronization in CPUs

©Simplilearn. All rights reserved.

Operations Performed on a Queue



Rear

Front

Enqueue

10 | 20 | 30 | 40 | 50

Dequeue

Queue

©Simplilearn. All rights reserved.

Queue

Duration: 20 min.

Problem Statement:
Write a program to demonstrate working of a queue.

ASSISTED PRACTICE

©Simplilearn. All rights reserved.

Assisted Practice: Guidelines

Steps to demonstrate the functionality of queue:

1. Create a Java project in your IDE
2. Write a program in Java to create a queue
3. Initialize the .git file
4. Add and commit the program files
5. Push the code to your GitHub repositories

©Simplilearn. All rights reserved.

Key Takeaways

- Data structure is a way of effectively organizing data.
- An array is a collection of elements of like data types stored at contiguous memory locations. The first element refers to index 0.
- Several operations can be performed on arrays like array rotations, order statistics, range queries, and matrix.
- Linked list is a collection of objects called nodes that contain two fields, a data field and a pointer to the address of the next node in the memory.
- Stacks are called Last-In-First-Out lists. Operations like addition and deletion are performed at one end called the **top**.
- Queues are known as First-In-Last-Out lists.

©Simplilearn. All rights reserved.

Longest Increasing Subsequence

Duration: 30 min.

Problem Statement:
As a developer, you are given a project to find out the longest increasing subsequence.

LESSON-END PROJECT

©Simplilearn. All rights reserved.