Data Structures & Algorithms

What is Data?

1

Data are the raw facts, unorganized facts that need to be processed. Data can be something simple and seemingly random and useless until it is organized.

What is Information?

When data is processed, organized, structured or presented in a given context so as to make it useful, it is called information.

Difference between Data & Information

BASIS FOR COMPARISON	DATA	INFORMATION
Meaning	Data means raw facts gathered about someone or something, which is bare and random.	Facts, concerning a particular event or subject, which are refined by processing is called information.
What is it?	It is just text and numbers.	It is refined data.
Based on	Records and Observations	Analysis
Form	Unorganized	Organized
Useful	May or may not be useful.	Always
Specific	No	Yes
Dependency	Does not depend on information.	Without data, information cannot be processed.

Algorithm, Program & Data Structure?

Algorithm Outline, the essence of a computational procedure, step by step instructions

Program An Implementation of Algorithm in some programming language

Data Structure Organization of data needed to solve the problem

Algorithm V/s Pseudocode

Algorithm	Pseudocode
Systematic logical approach which is a well-defined, step-by-step procedure that allows a computer to solve a problem.	It is one of the methods which can be used to represent an algorithm for a program.
Algorithms can be expressed using natural language, flowcharts, etc.	Pseudocode allows you to include several control structures such as While, If-then-else, Repeat-until, for and case, which is present in many high-level languages.

Data Structure & Algorithm

Types of Data Structure

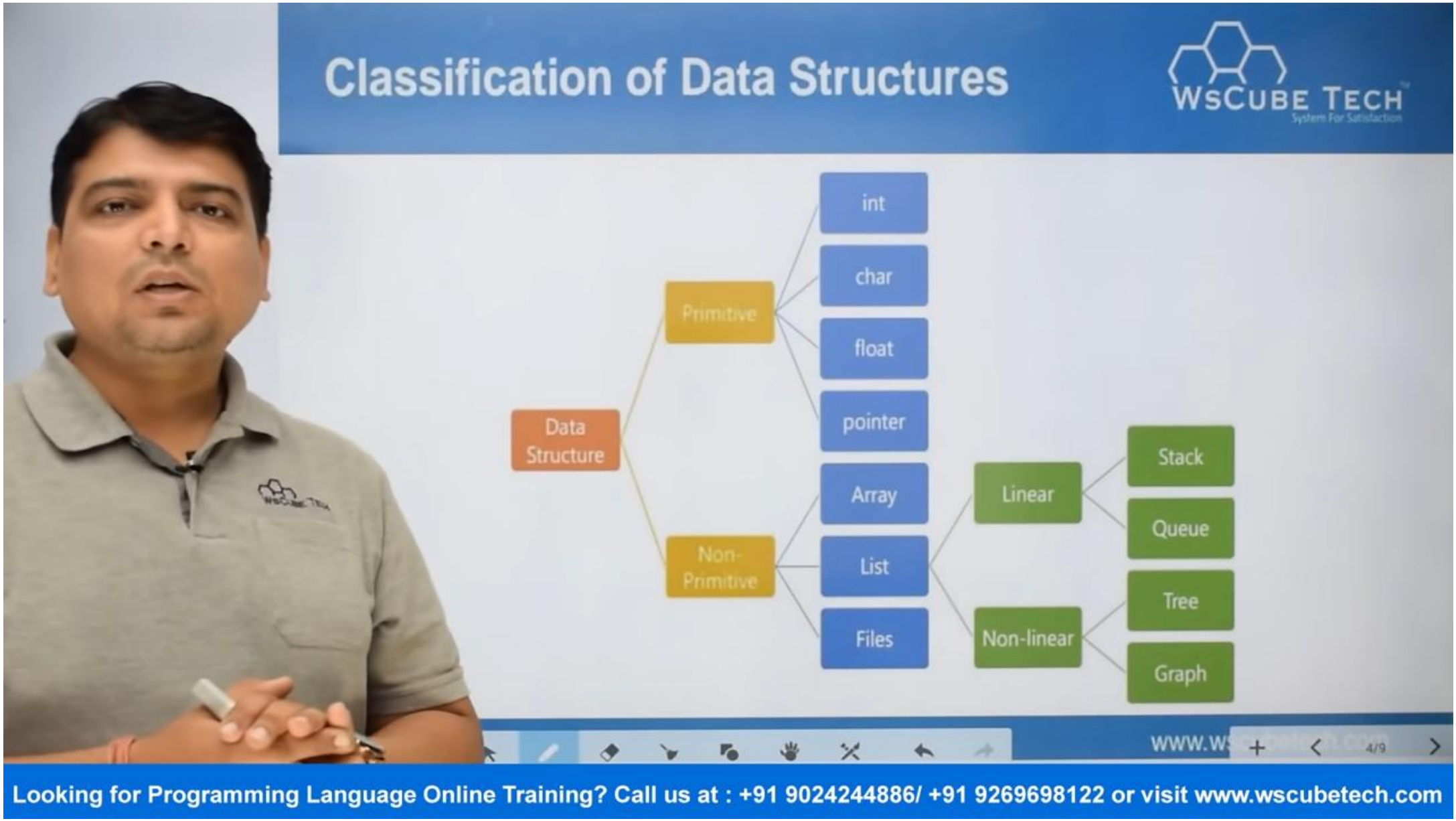
Data Structures

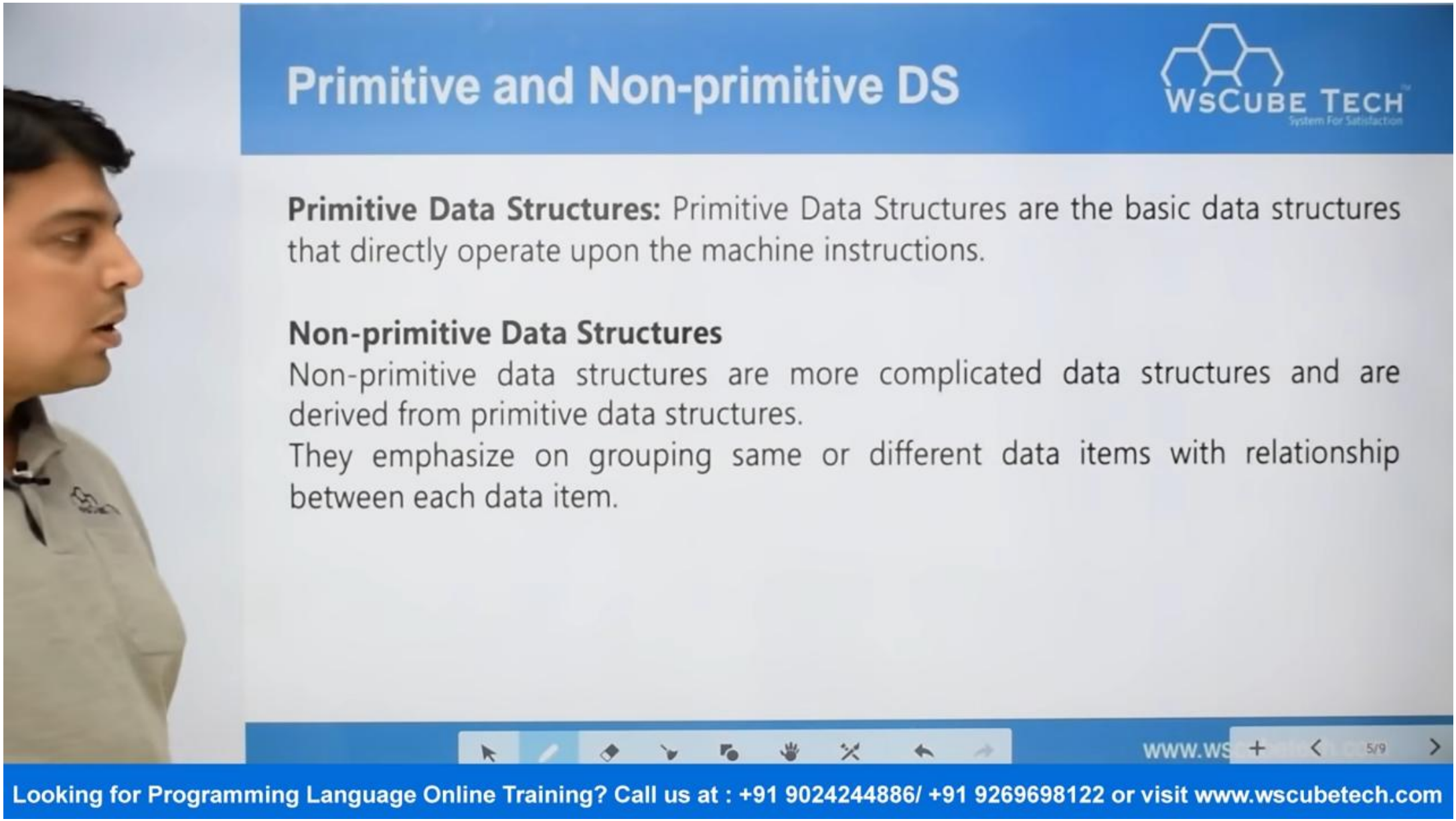
WsCube Tech
System For Satisfaction

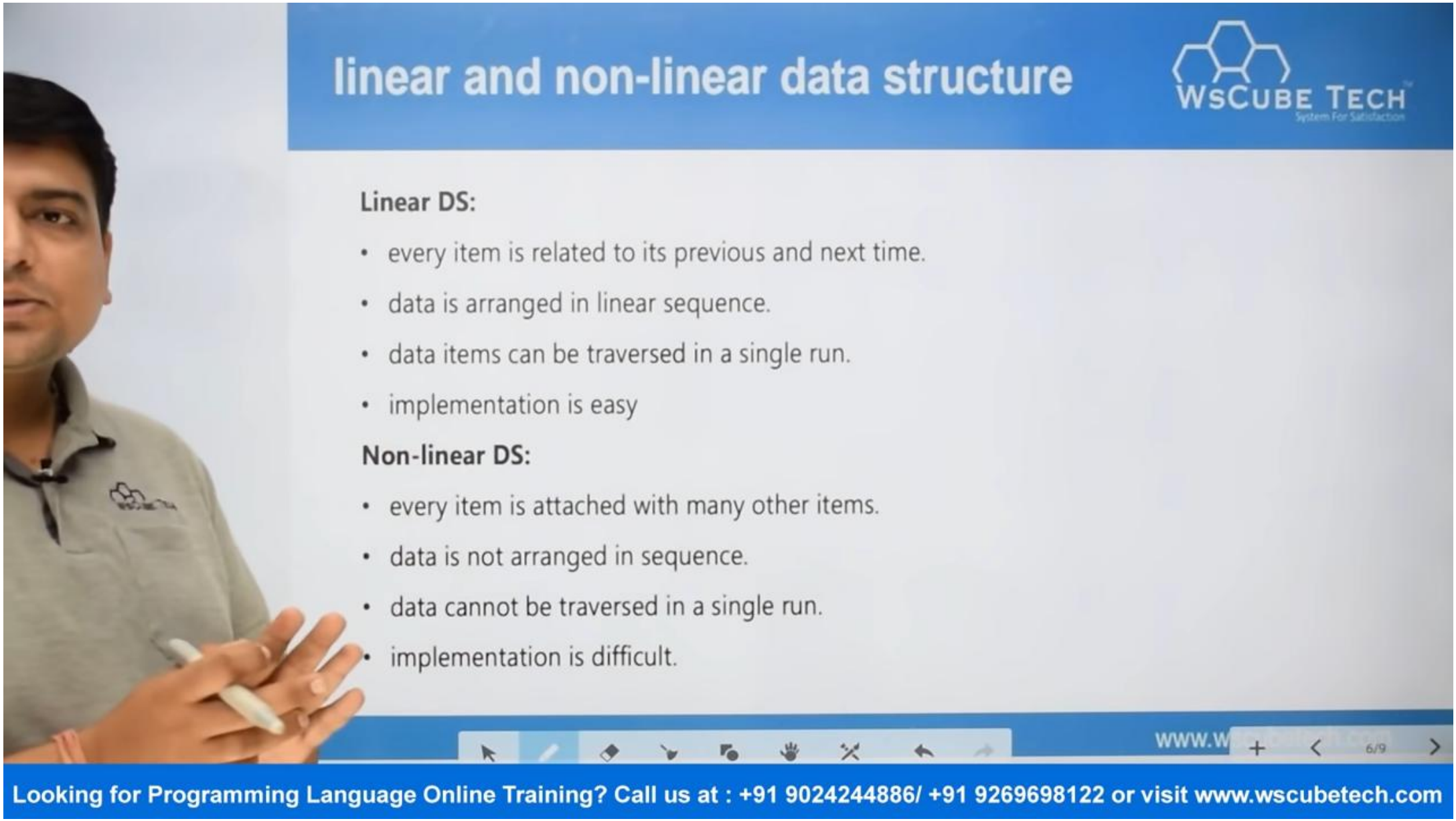
Data Structure is a way of collecting and organizing data in such a way that we can perform operations on these data in an effective way.

Types Of Data Structures

- Primitive data structures
- Non-primitive data structure



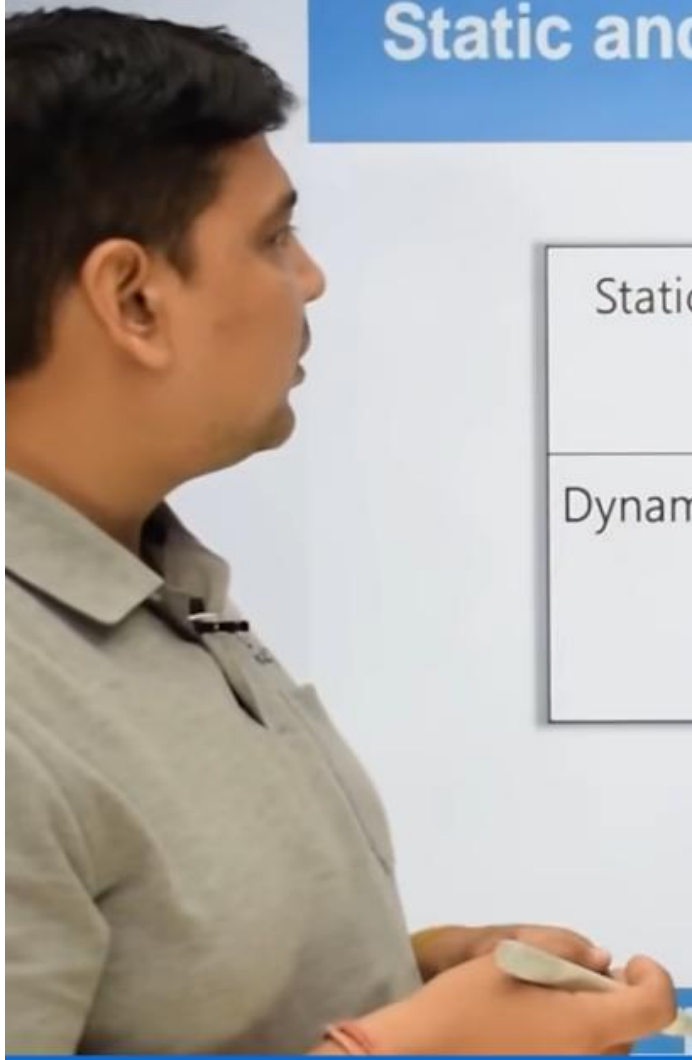




Static and Dynamic DS

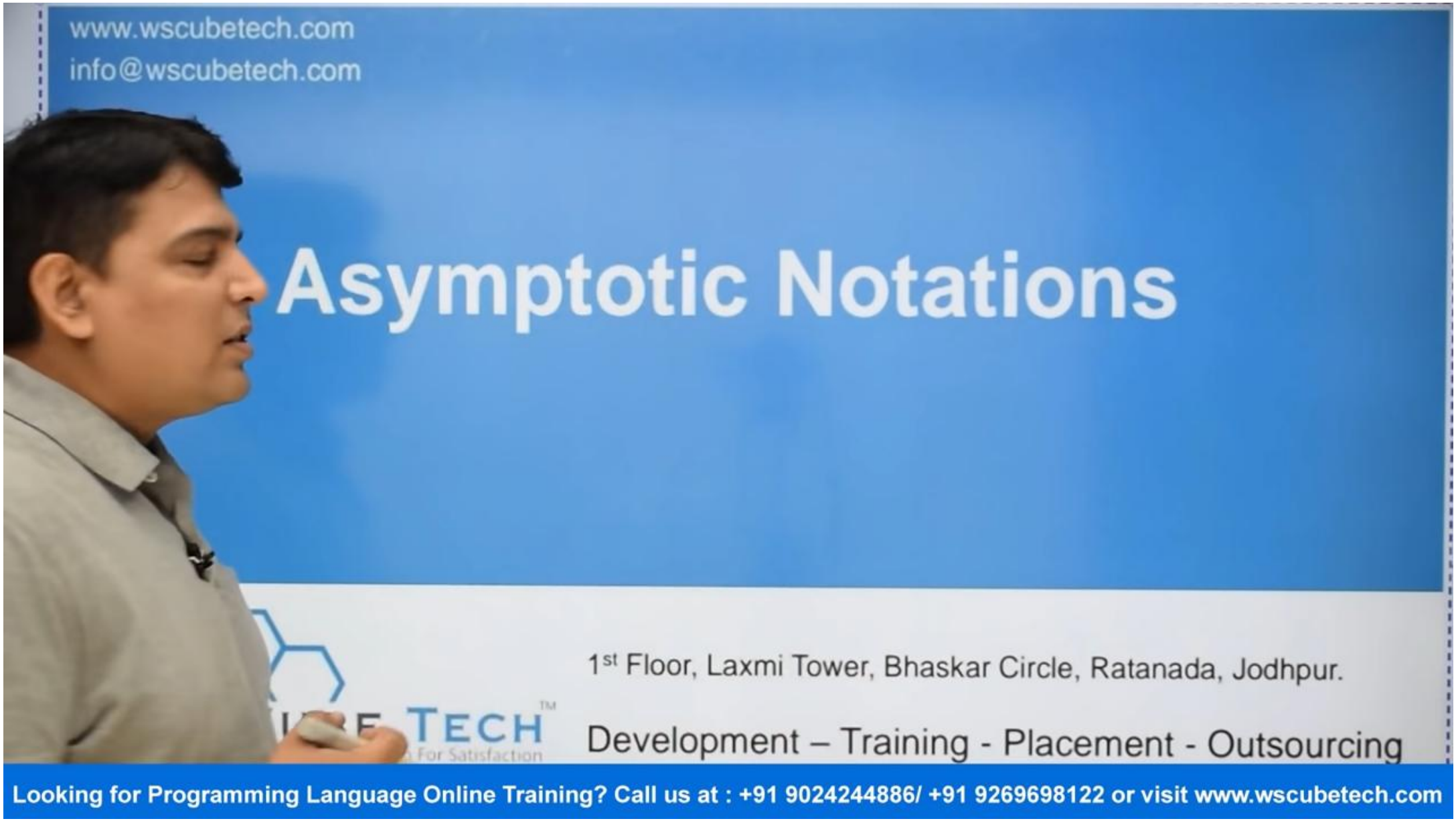
WSCUBE TECH
System For Satisfaction

Static	Static data structures are those whose sizes and structures associated memory locations are fixed, at compile time. Example: **Array**
Dynamic	Dynamic structures are those which expands or shrinks depending upon the program need and its execution. Also, their associated memory locations changes. Example: **Linked List created using pointers**



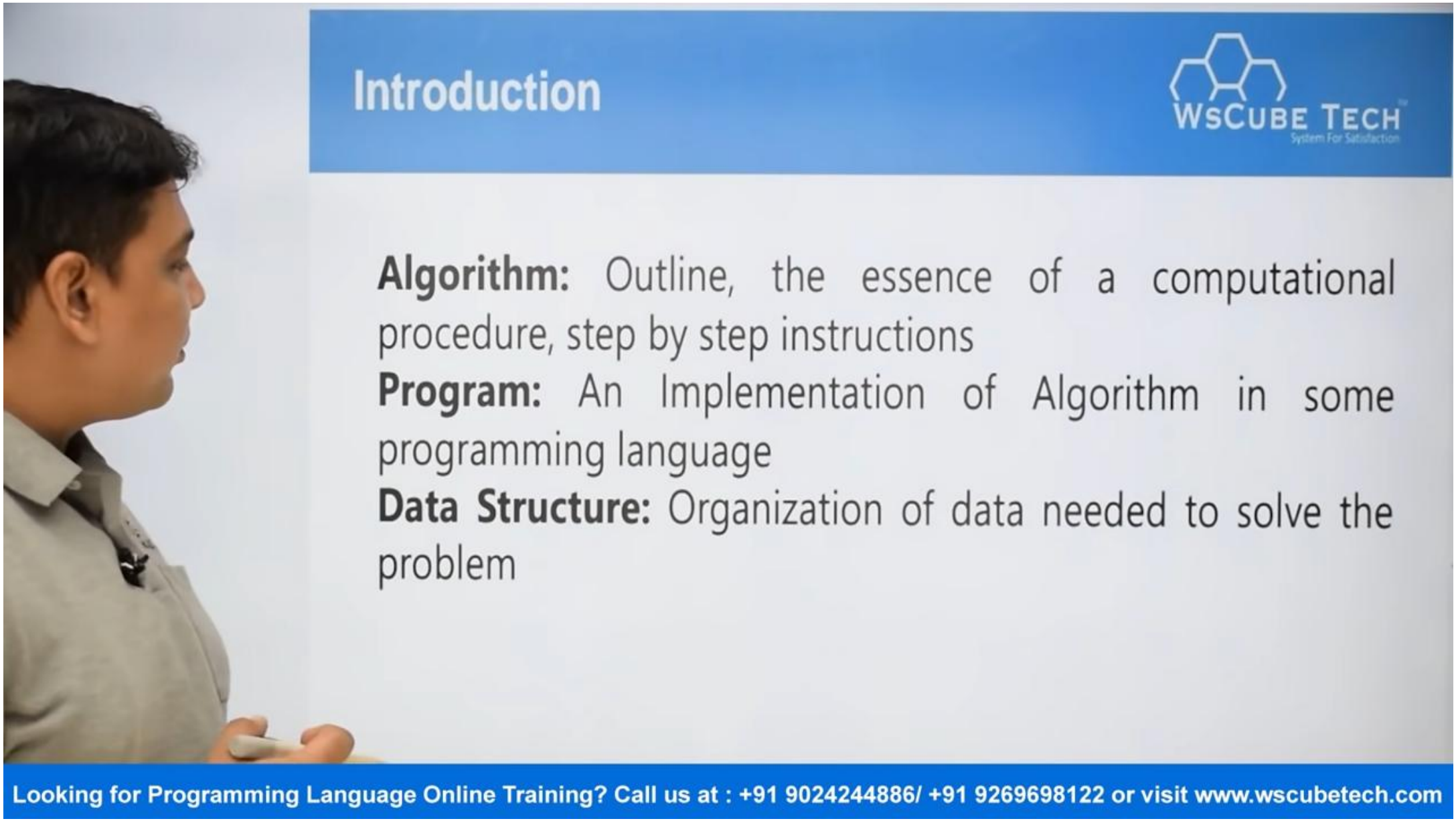
Homogeneous and Non-Homogeneous DS

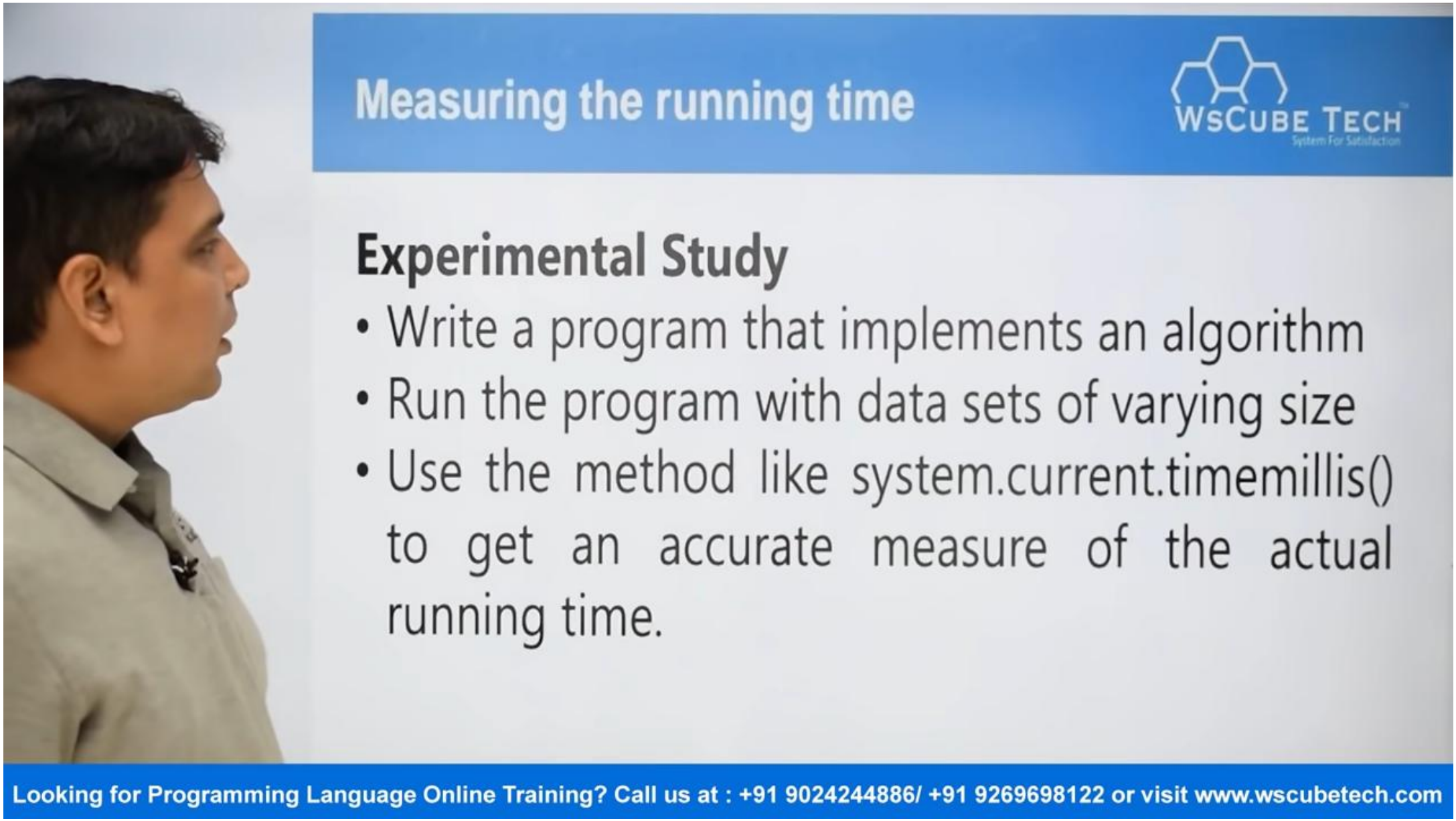
Homogeneous	In homogeneous data structures, all the elements are of same type. Example: **Array**
Non-Homogeneous	In Non-Homogeneous data structure, the elements may or may not be of the same type. Example: **Structures**

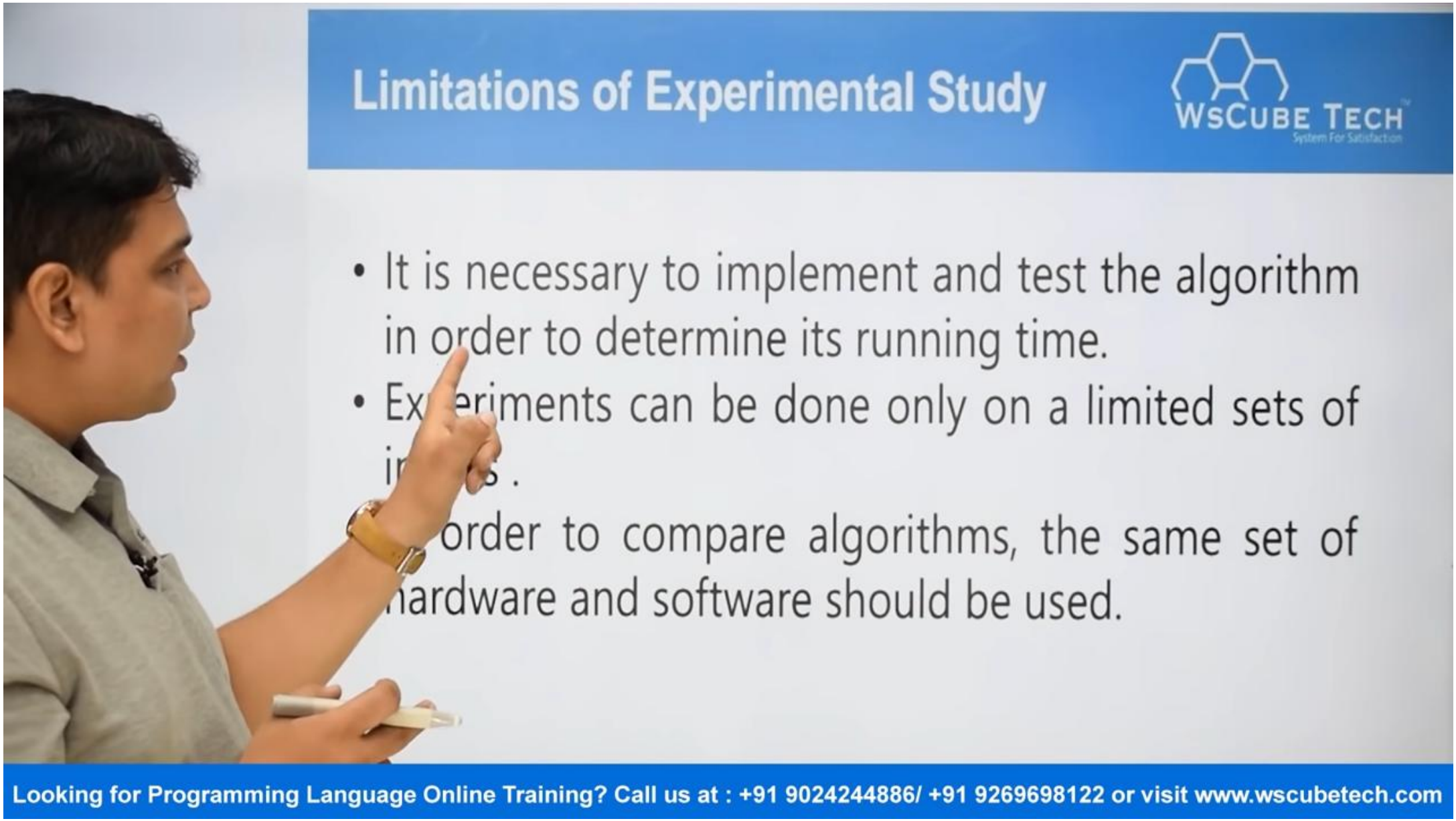


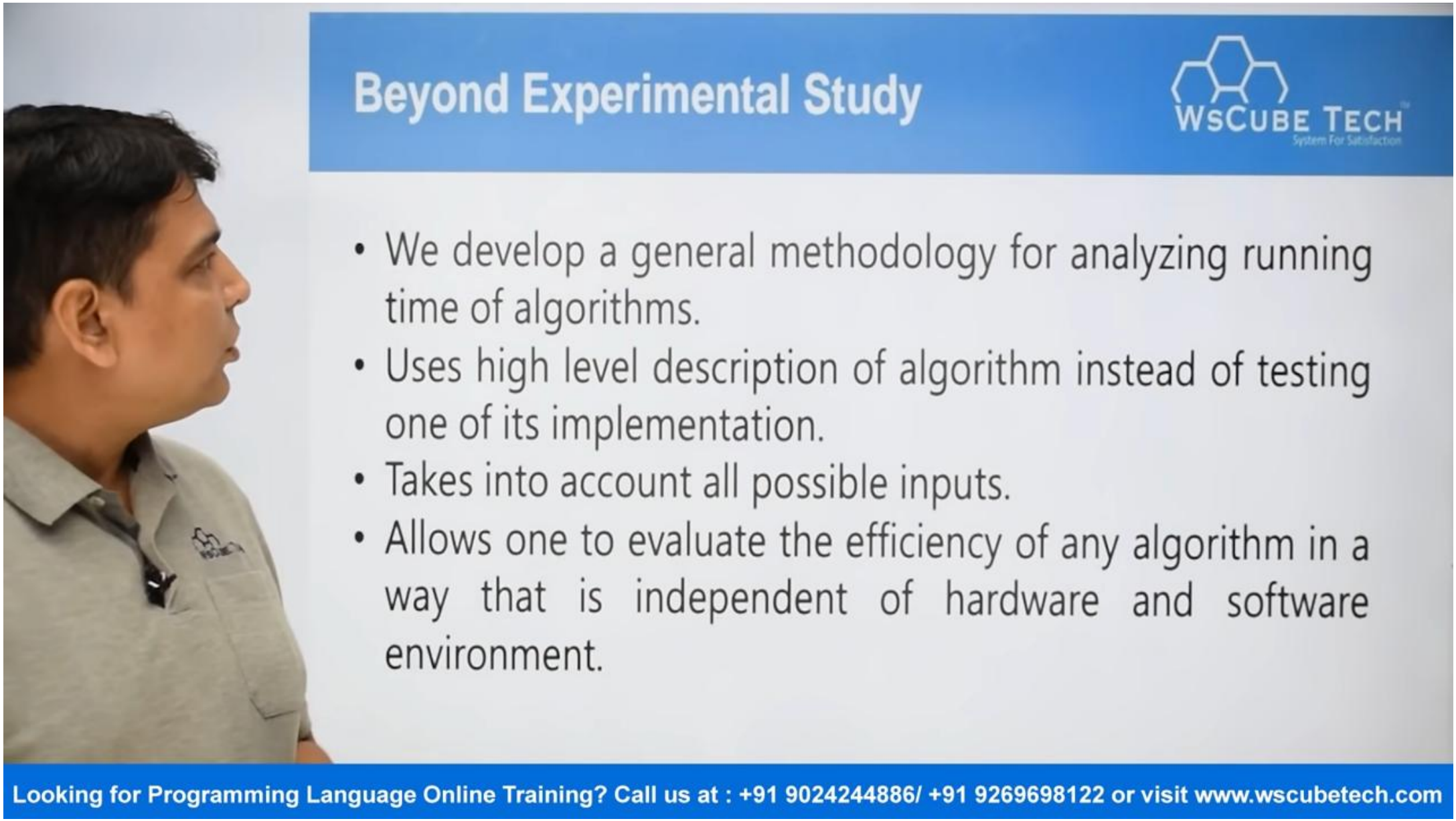
Data Structure & Algorithm

Asymptotic Notation

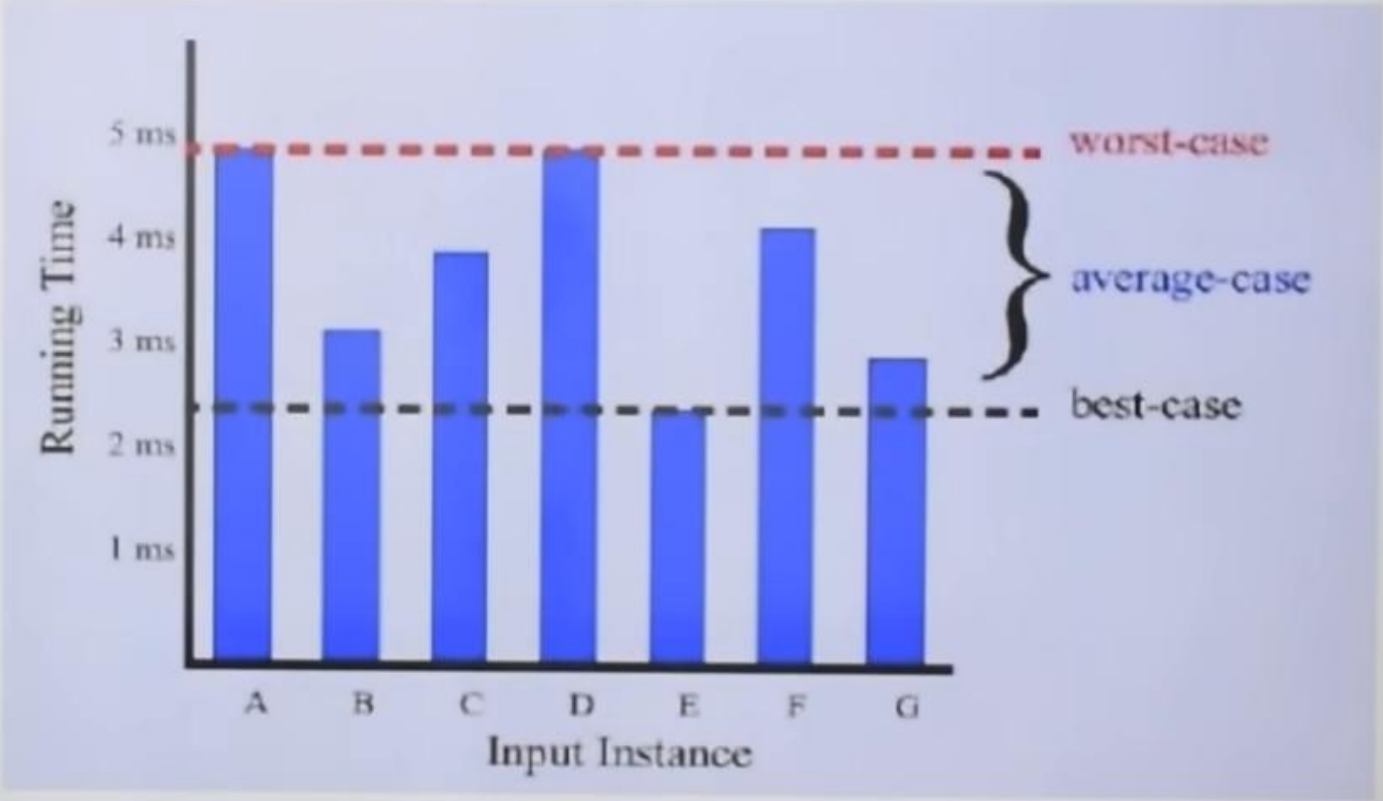


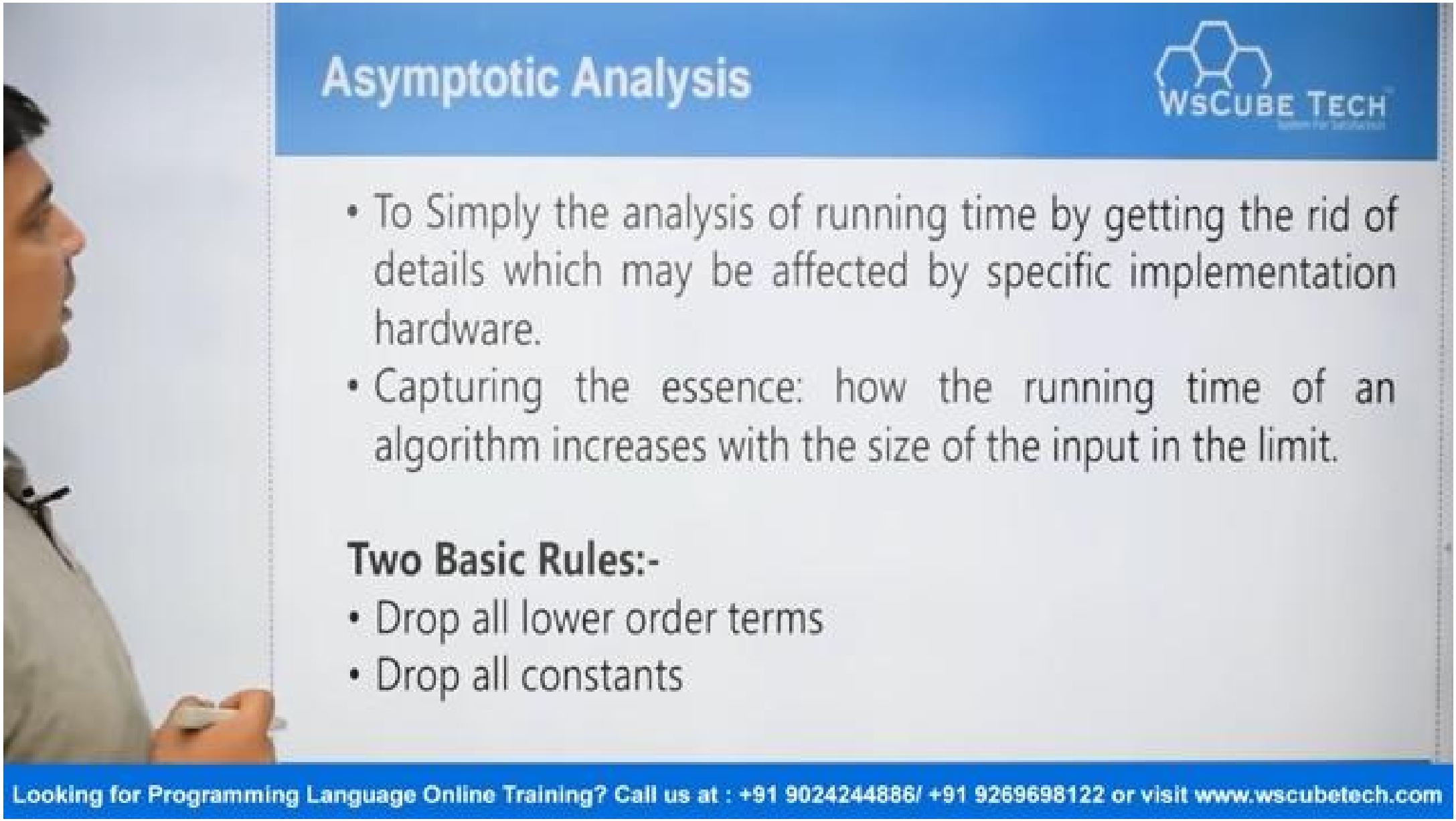


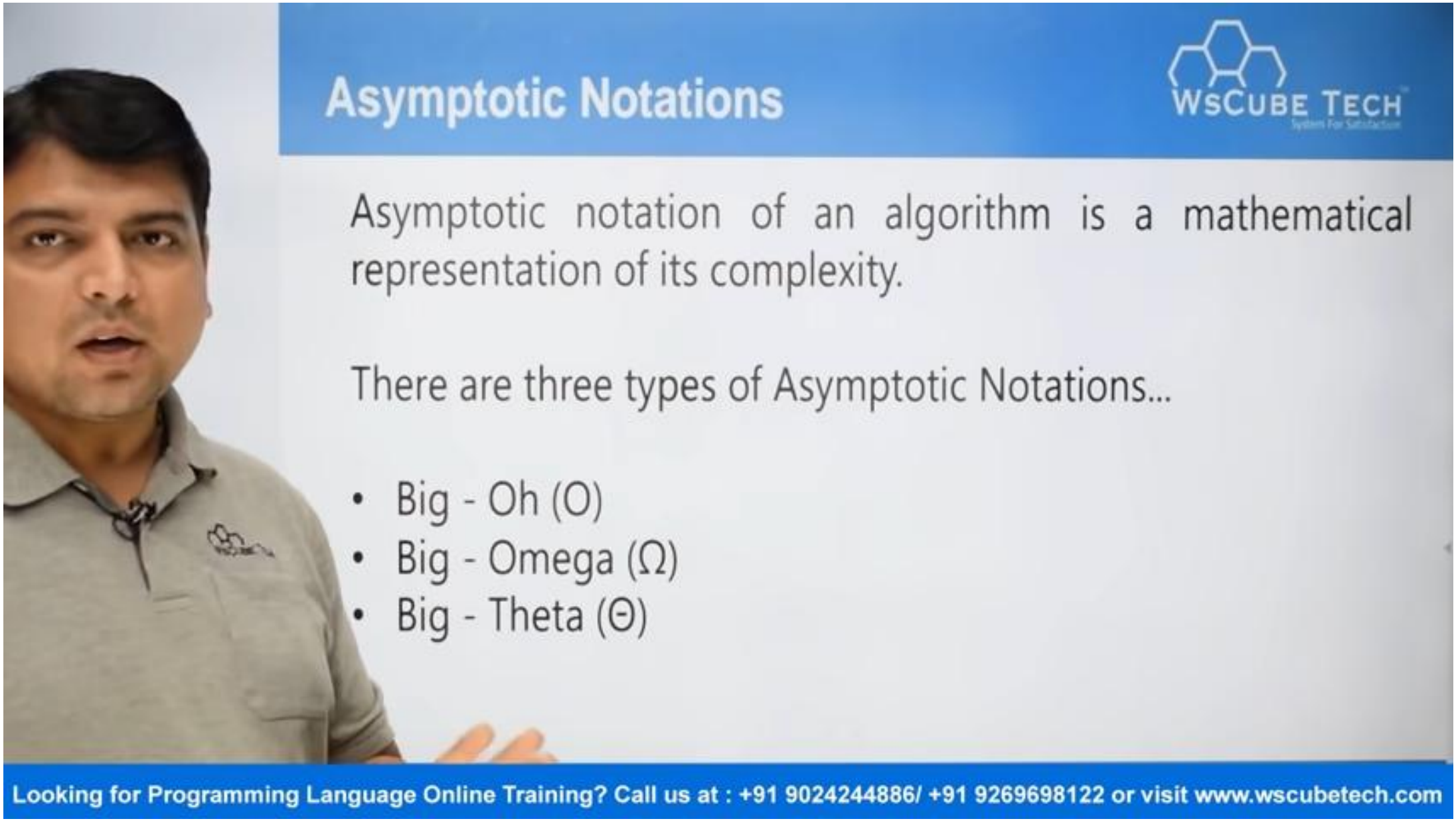


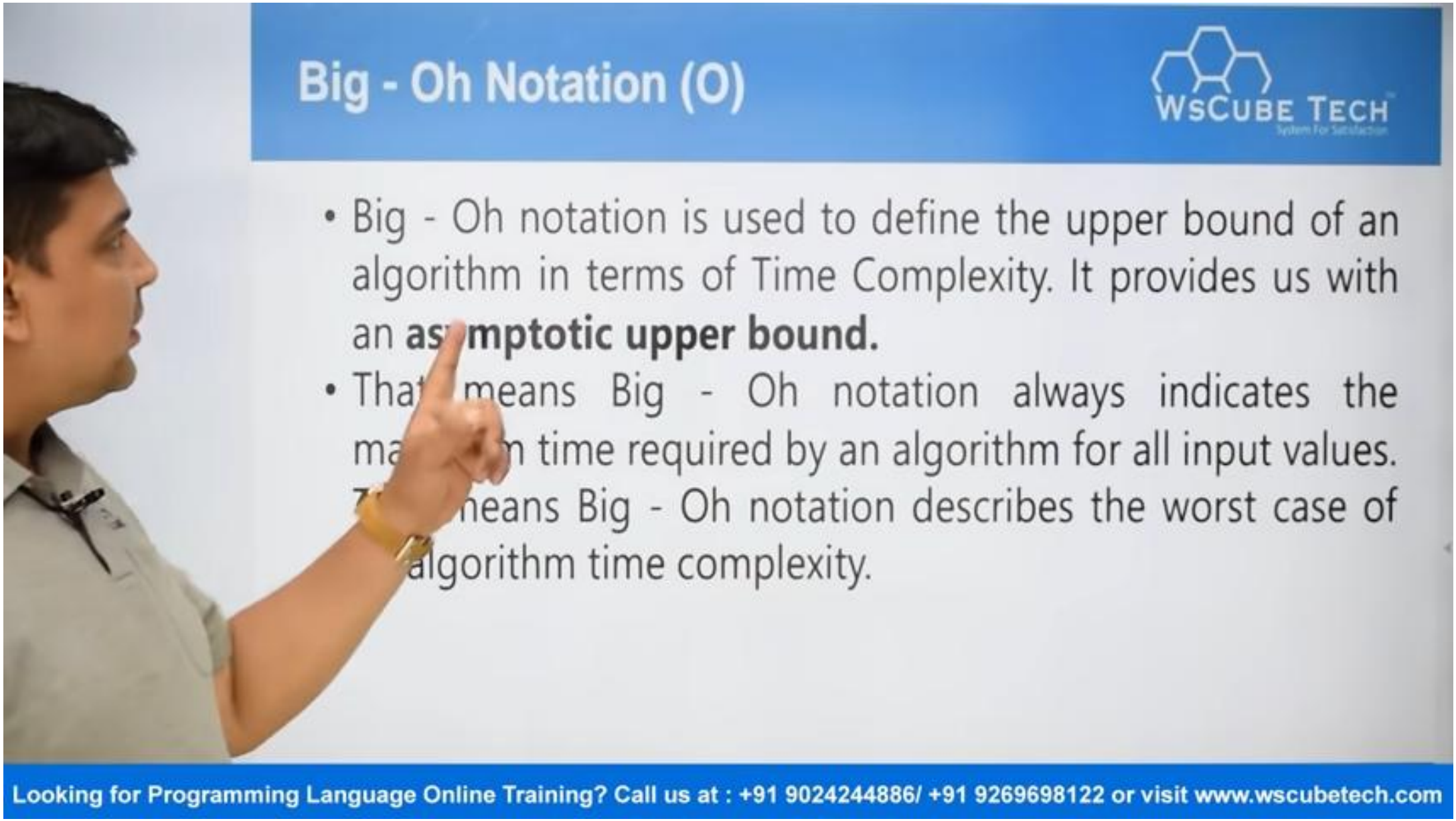


Best/Worst/Average Case





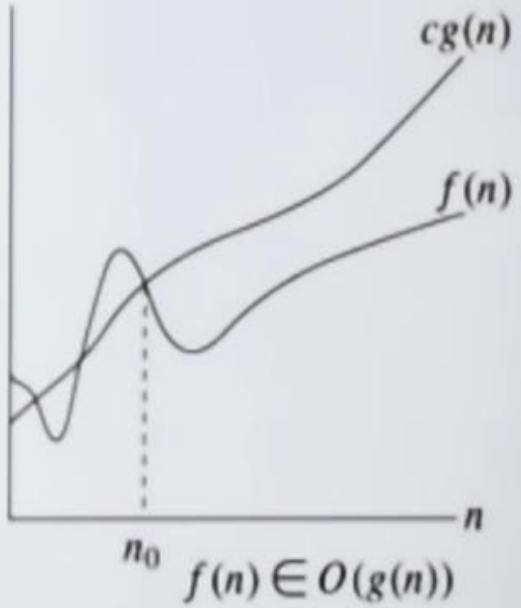

Asymptotic Notations
WSCUBE TECH
Asymptotic notation of an algorithm is a mathematical representation of its complexity.
There are three types of Asymptotic Notations...
• Big - Oh (O)
• Big - Omega (Ω)
• Big - Theta (Θ)
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

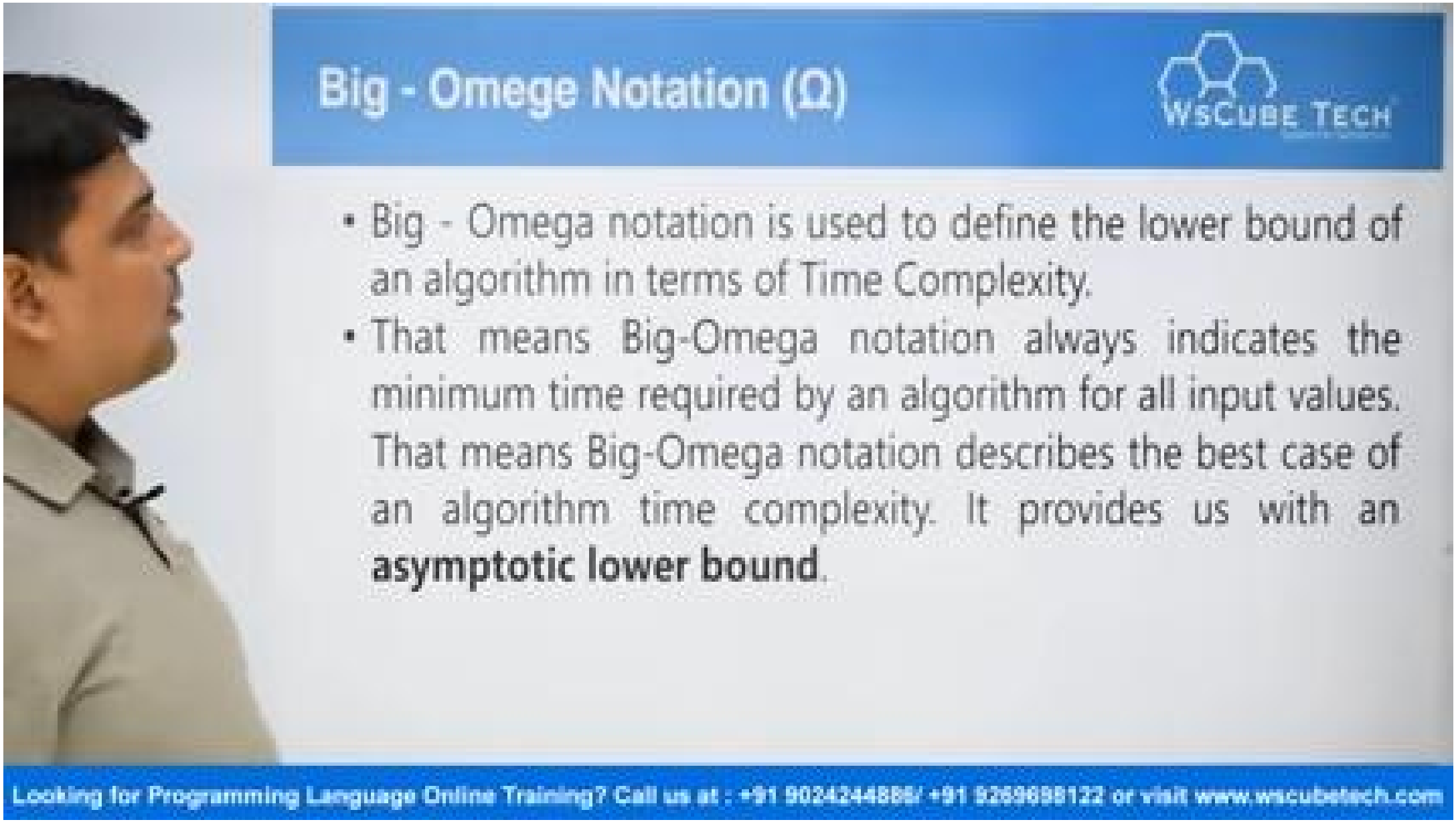


Big - Oh Notation (O)

WsCube Tech
System For Satisfaction

- f(n) is your algorithm runtime, and g(n) is an arbitrary time complexity you are trying to relate to your algorithm. f(n) is O(g(n)), if for some real constants c (c > 0) and n0, f(n) <= c g(n) for every input size n (n > n0).

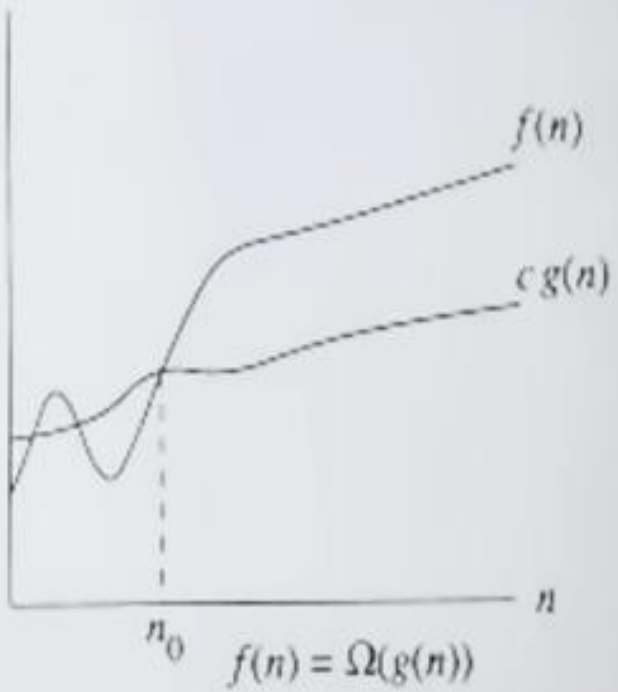


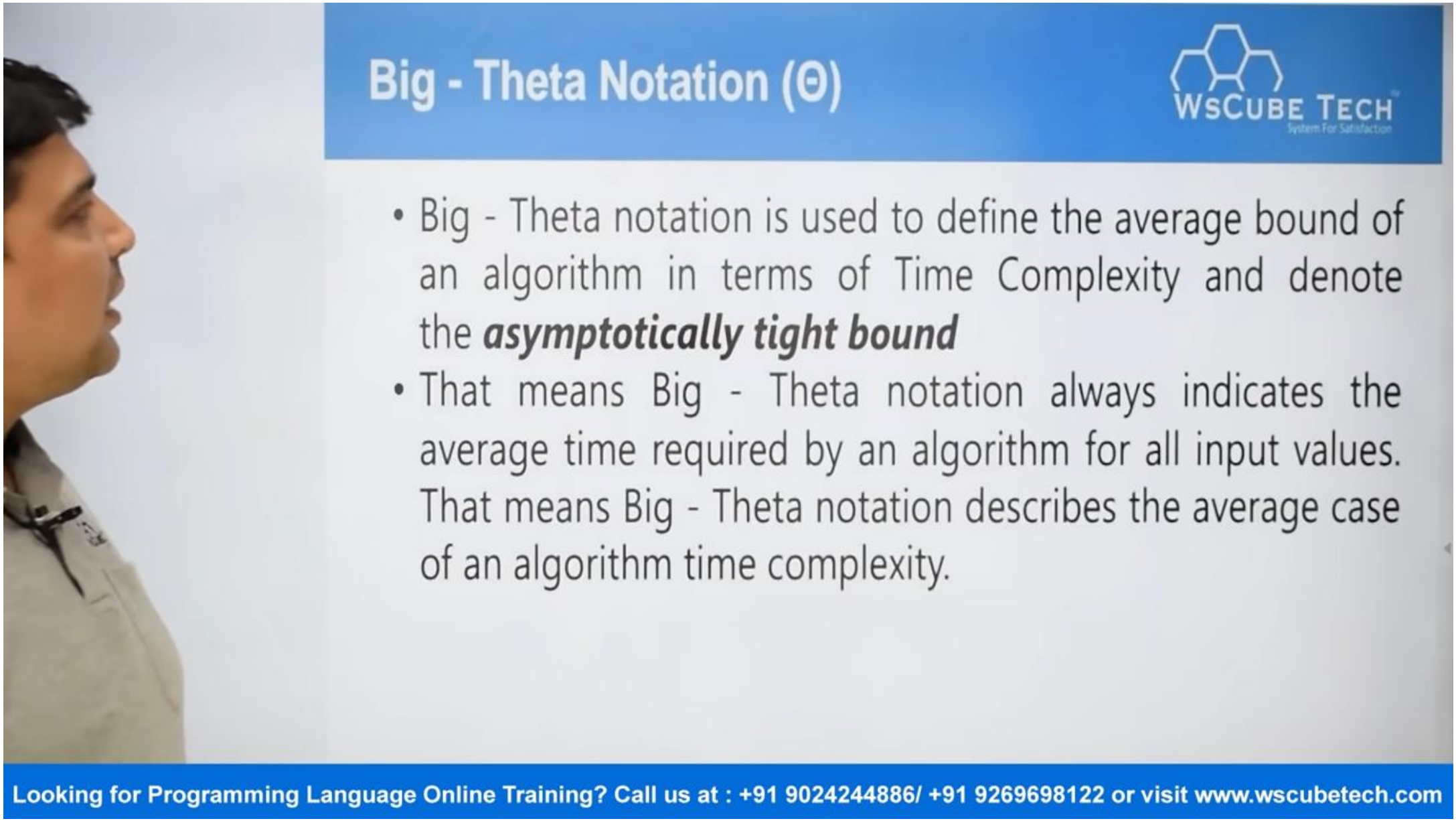

Big - Omege Notation (Ω)
WSCUBE TECH
• Big - Omega notation is used to define the lower bound of an algorithm in terms of Time Complexity.
• That means Big-Omega notation always indicates the minimum time required by an algorithm for all input values. That means Big-Omega notation describes the best case of an algorithm time complexity. It provides us with an asymptotic lower bound.
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

Big - Omege Notation (Ω)

WSCUBE TECH

- f(n) is Ω(g(n)), if for some real constants c (c > 0) and n0 (n0 > 0), f(n) is >= c g(n) for every input size n (n > n0).

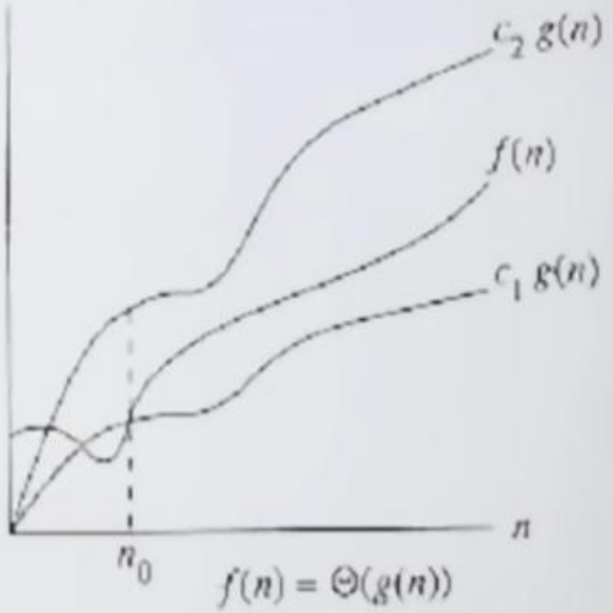


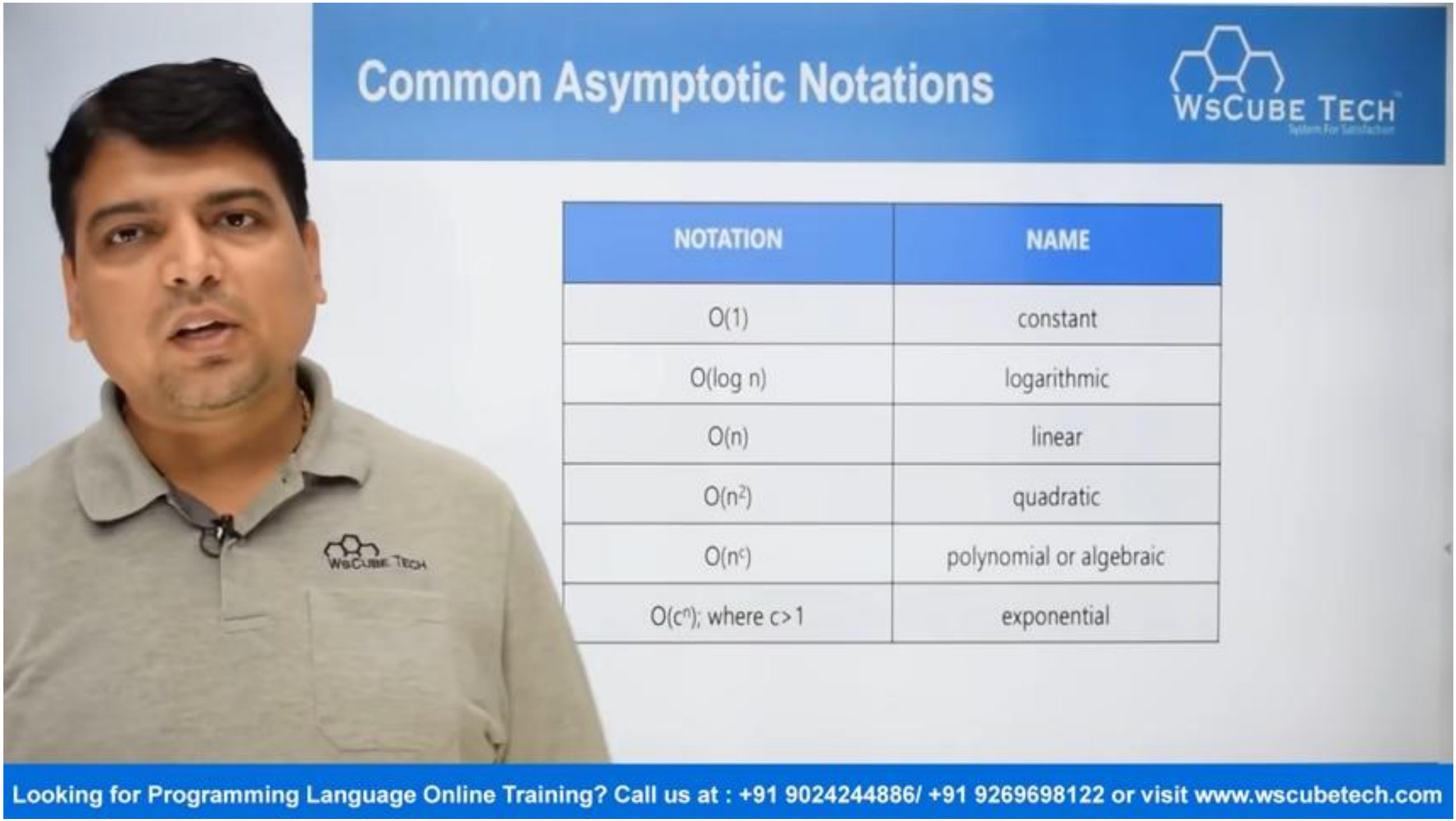


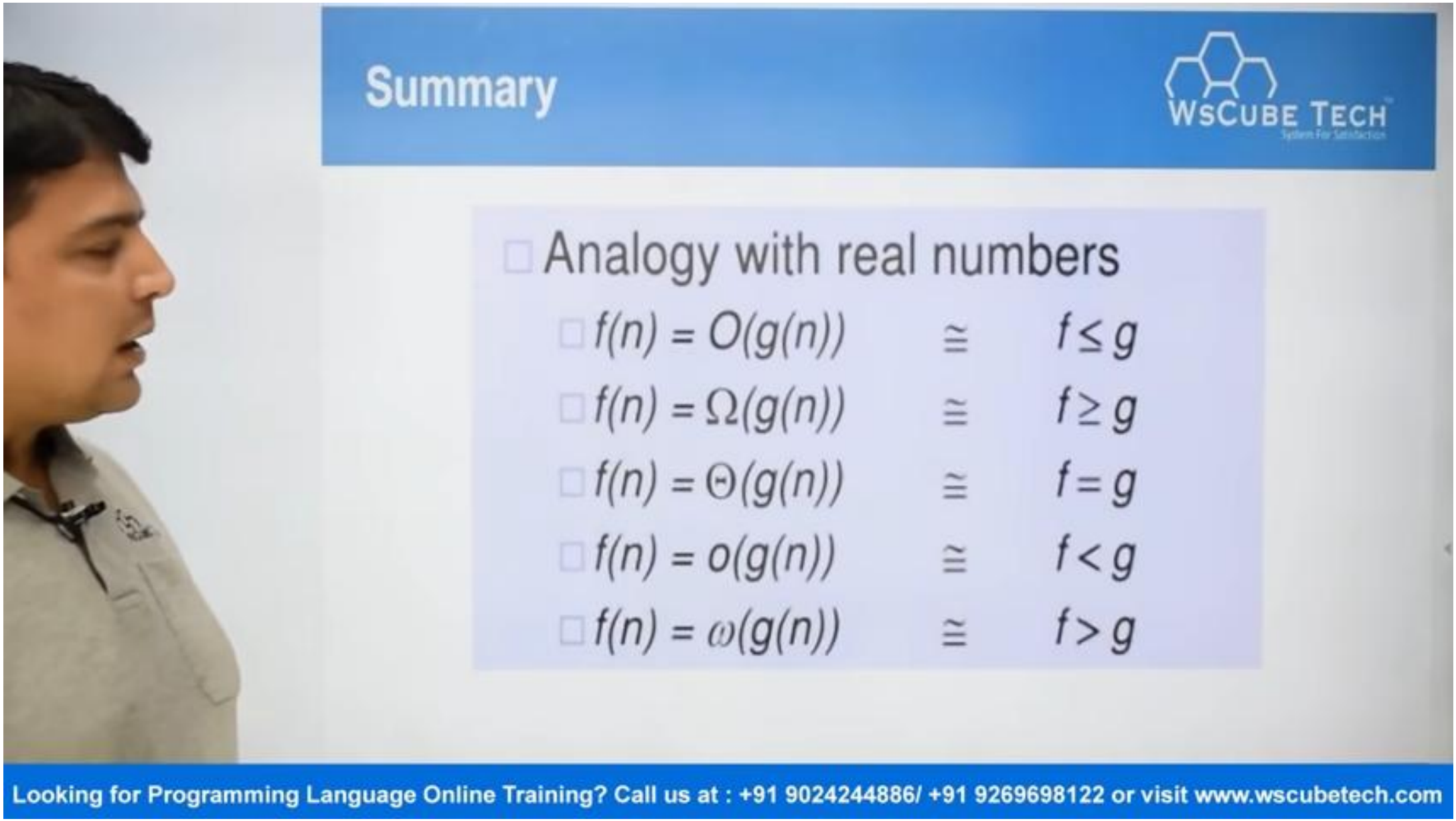
Big - Theta Notation (Θ)

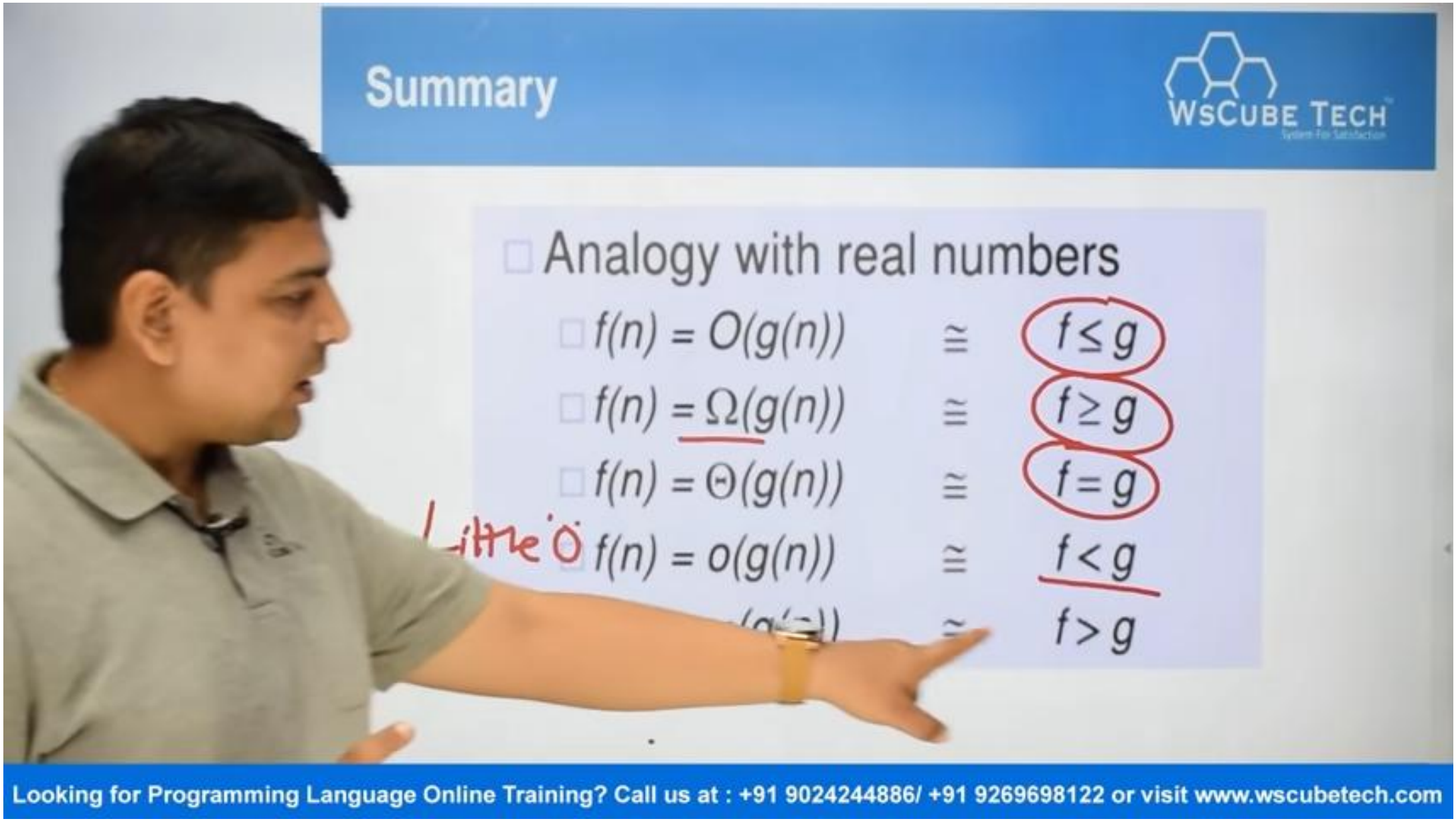
WsCube Tech
System For Satisfaction

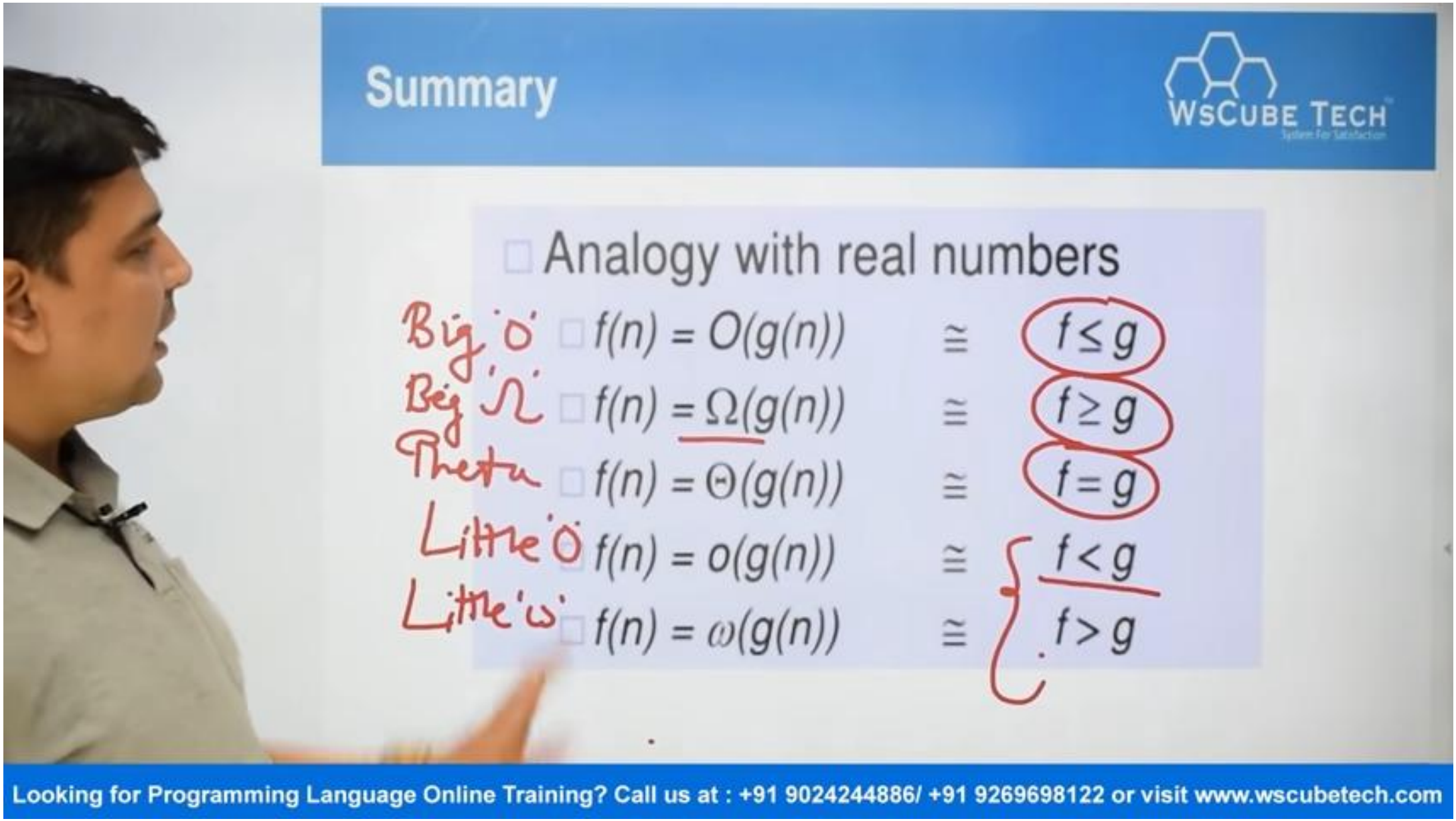
- function f(n) as time complexity of an algorithm and g(n) is the most significant term. If C1 g(n) <= f(n) <= C2 g(n) for all n >= n0, C1 > 0, C2 > 0 and n0 >= 1. Then we can represent f(n) as Θ(g(n)).




Common Asymptotic Notations
WsCube Tech
NOTATION
NAME
O(1)
constant
O(log n)
logarithmic
O(n)
linear
O(n²)
quadratic
O(nᶜ)
polynomial or algebraic
O(cⁿ); where c>1
exponential
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com





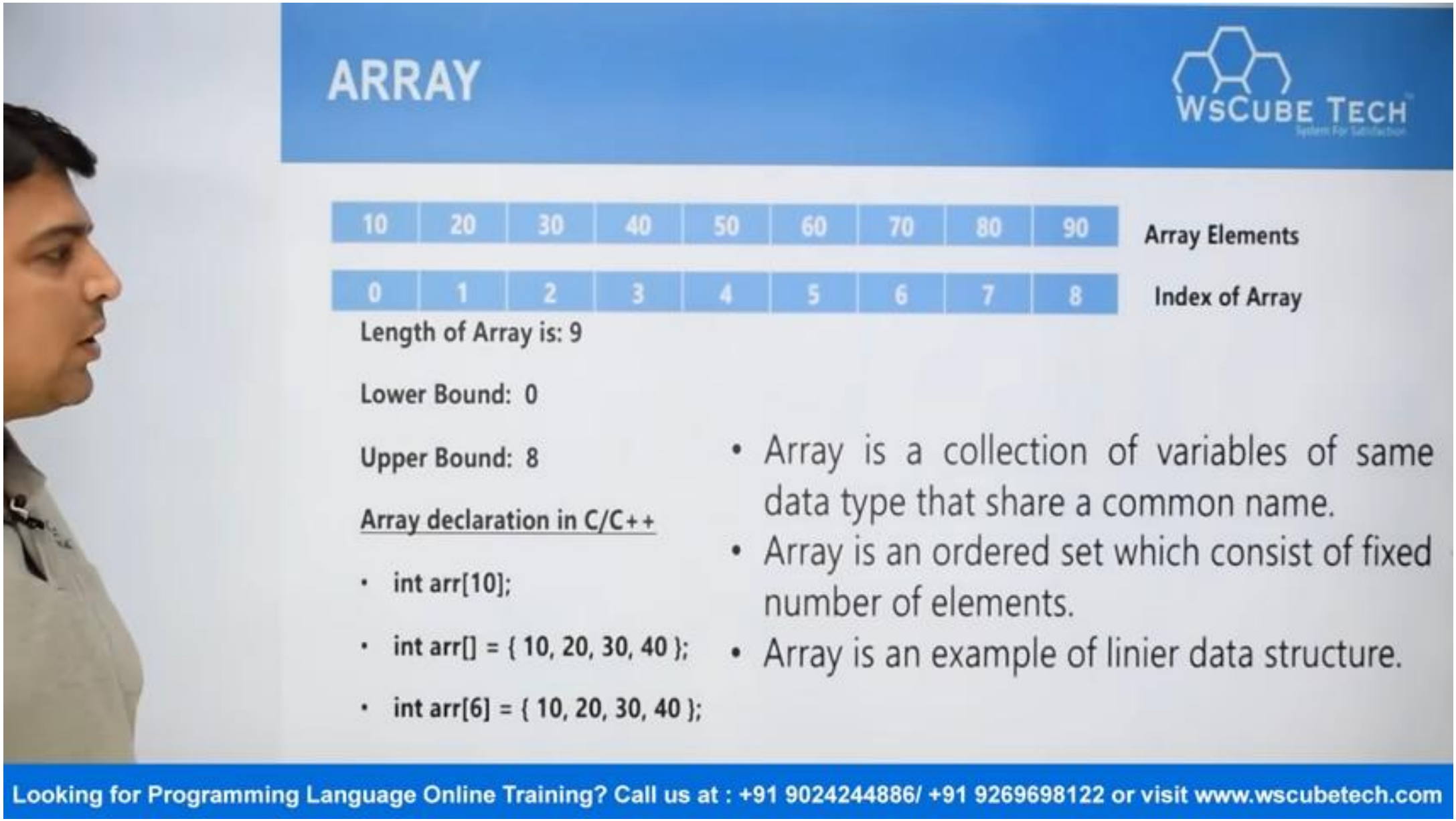


Data Structure & Algorithm

Array in DSA

WsCube Tech™
System For Satisfaction


www.wscubetech.com
Array in DSA
Laxmi Tower, Bhaskar Circle, Ratanada, Jodhpur.
ment – Training - Placement - Outsourcing
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com



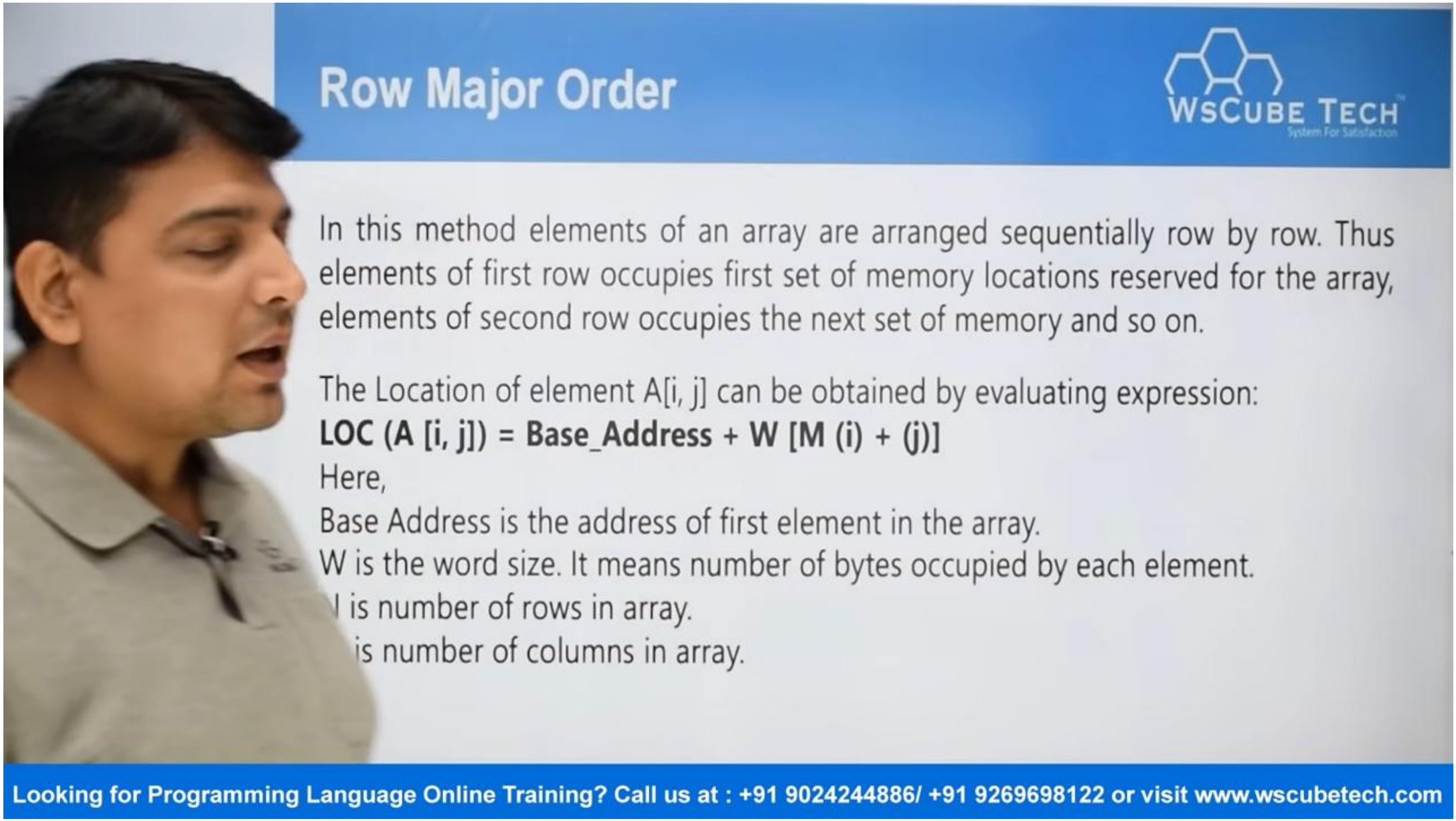
ARRAY

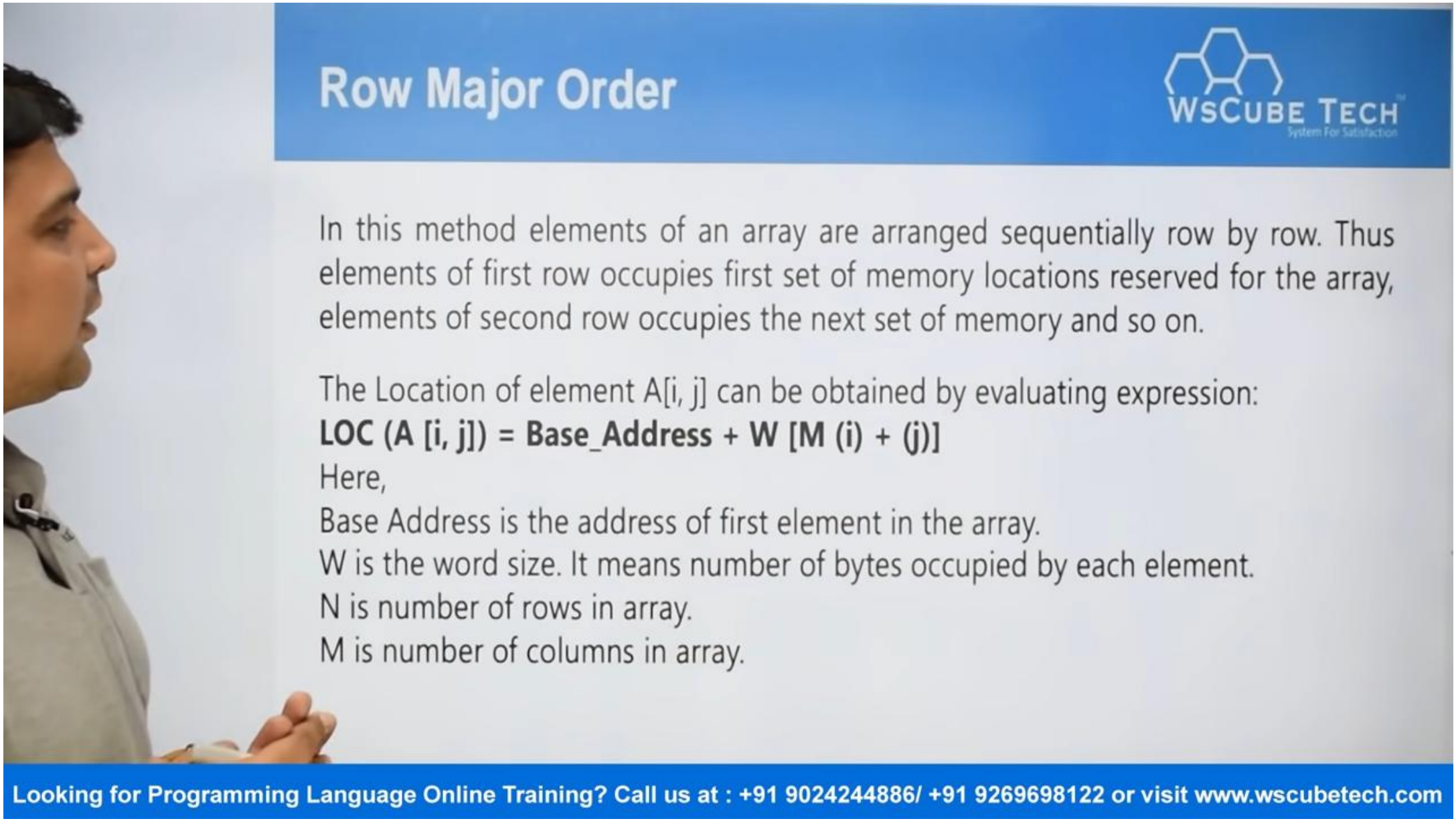
WSCUBE TECH

In One Dimensional Array location of element A[i] can be calculated using following equation:

LOC (A[i]) = Base_Address + W * (i)

Base_Address is the address of first element in the array.
W is the word size. It means number of bytes occupied by each element.





Example of Row Major Order

WSCUBE TECH
System For Satisfaction

11 Arr[0][0] 1000	**12** Arr[0][1] 1002	**13** Arr[0][2] 1004	**14** Arr[0][3] 1006
21 Arr[1][0] 1008	**22** Arr[1][1] 1010	**23** Arr[1][2] 1012	**24** Arr[1][3] 1014
31 Arr[2][0] 1016	**32** Arr[2][1] 1018	**33** Arr[2][2] 1020	**34** Arr[2][3] 1022

Suppose we want to calculate the address of element Arr[1, 2].

It can be calculated as follow:

Here,

Base Address = 1000, W= 2, M=4, N=3, i=1, j=2

LOC (A [i, j])=Base Address + W [M (i) + (j)]

LOC (A[1, 2])=1000 + 2 *[4*(1) + 2]

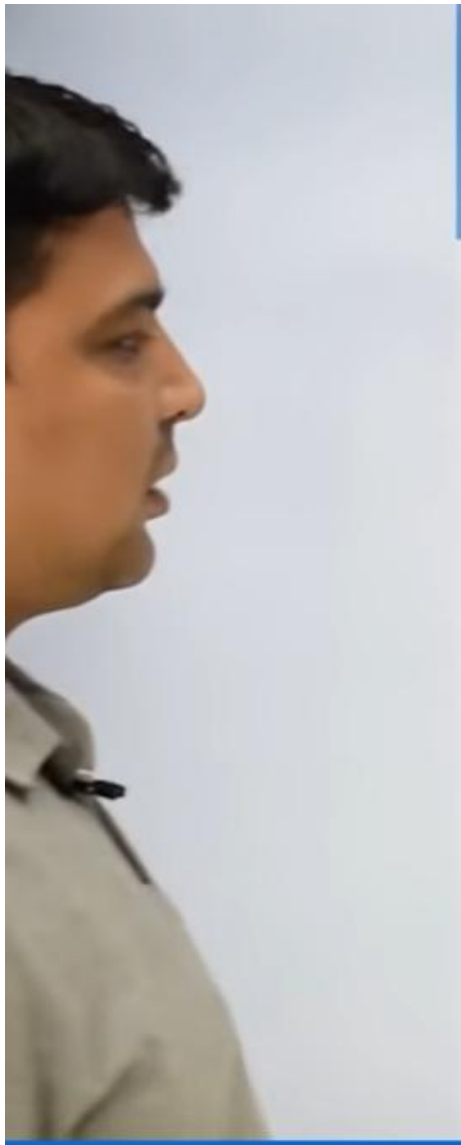
= 1000 + 2 * [4 + 2]

= 1000 + 2 * 6

= 1000 + 12

= **1012**

11	**12**	**13**	**14**	**21**	**22**	**23**	**24**	**31**	**32**	**33**	**34**



Column Major Order

WsCube Tech
System For Satisfaction

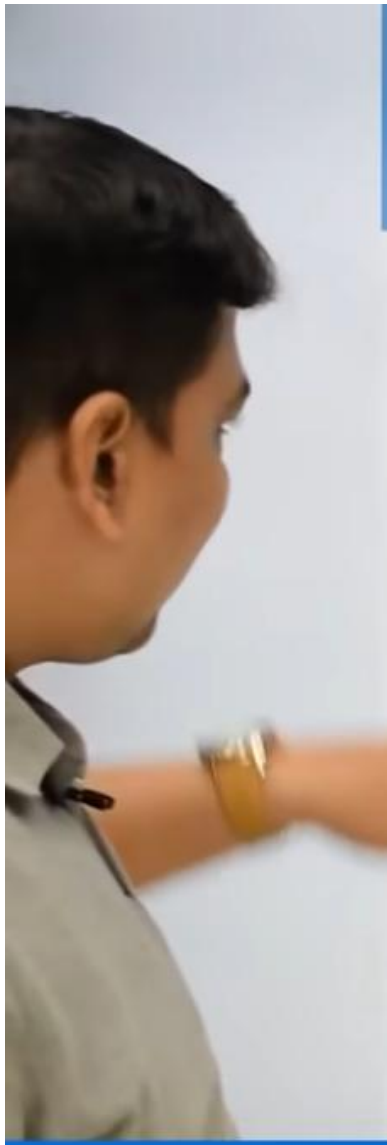
In this method elements of an array are arranged sequentially column by column. Thus elements of first column occupies first set of memory locations reserved for the array, elements of second column occupies the next set of memory and so on.

The Location of element A[i, j] can be obtained by evaluating expression:

LOC (A [i, j]) = Base_Address + W [N (j) + (i)]

Base_Address is the address of first element in the array.
W is the word size. It means number of bytes occupied by each element.
N is number of rows in array.
M is number of columns in array.

Example of Column Major Order



11 Arr[0][0] 1000	**12** Arr[0][1] 1006	**13** Arr[0][2] 1012	**14** Arr[0][3] 1018
21 Arr[1][0] 1002	**22** Arr[1][1] 1008	**23** Arr[1][2] 1014	**24** Arr[1][3] 1020
31 rr[2][0] 1004	**32** Arr[2][1] 1010	**33** Arr[2][2] 1016	**34** Arr[2][3] 1022

Suppose we want to calculate the address of element Arr[1, 2].

It can be calculated as follow:

Here,

Base Address = 1000, W= 2, M=4, N=3, i=1, j=2

LOC (A [i, j])=Base Address + W [N (j) + (i)]

LOC (A[1, 2])=1000 + 2 *[3*(2) + 1]

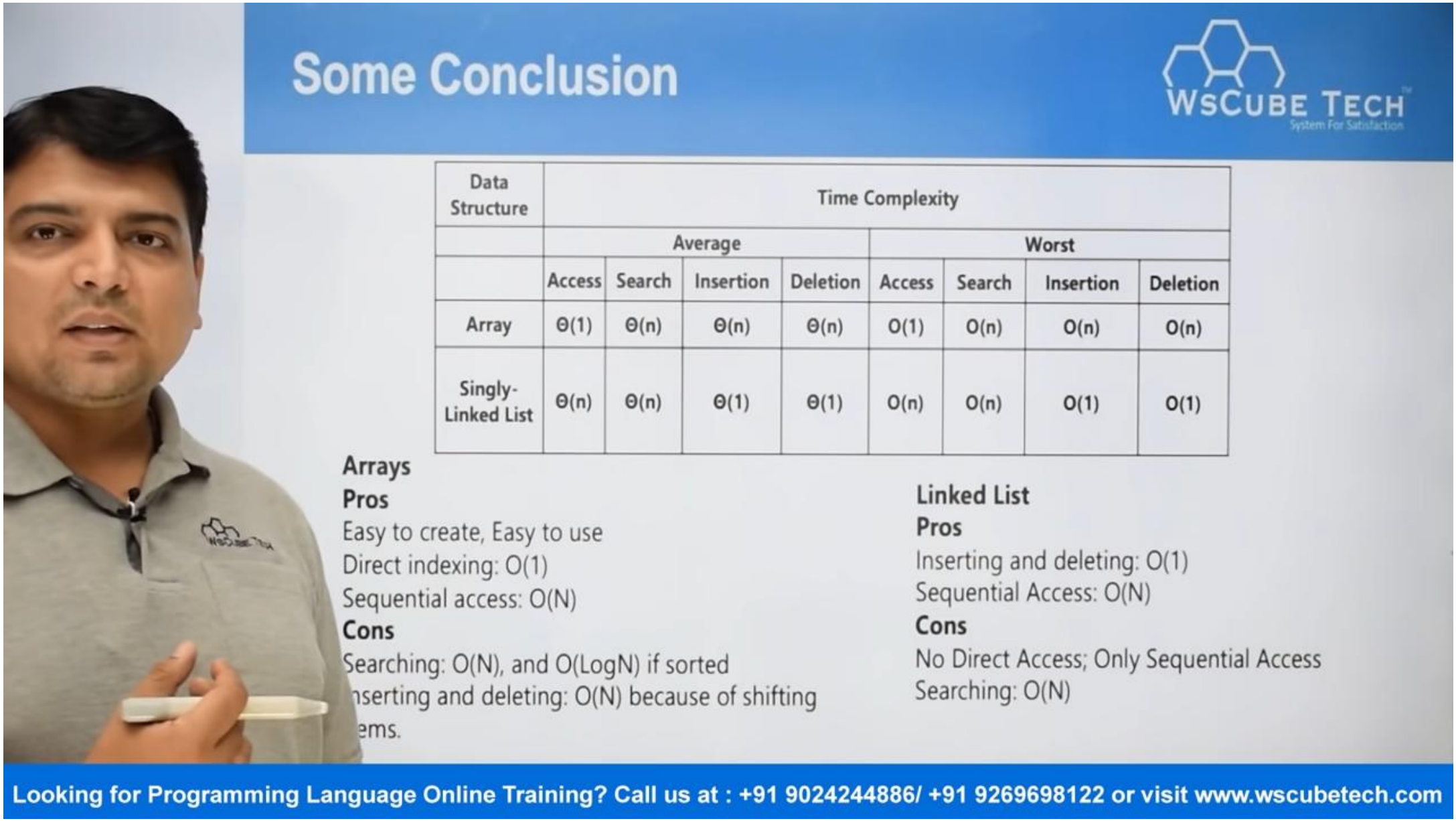
= 1000 + 2 * [6 + 1]

= 1000 + 2 * 7

= 1000 + 14

= **1014**

11	**21**	**31**	**12**	**22**	**32**	**13**	**23**	**33**	**14**	**24**	**34**

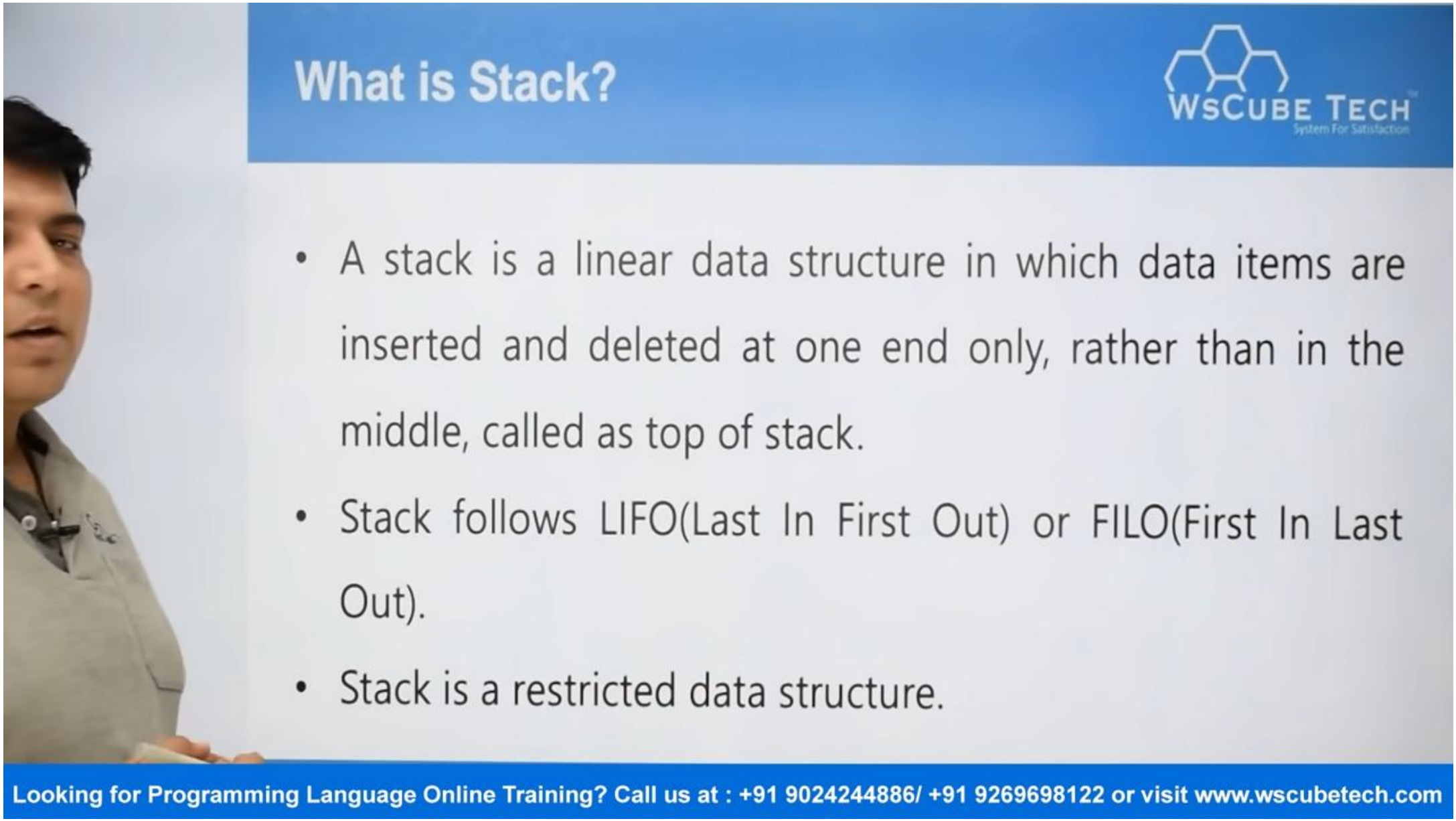


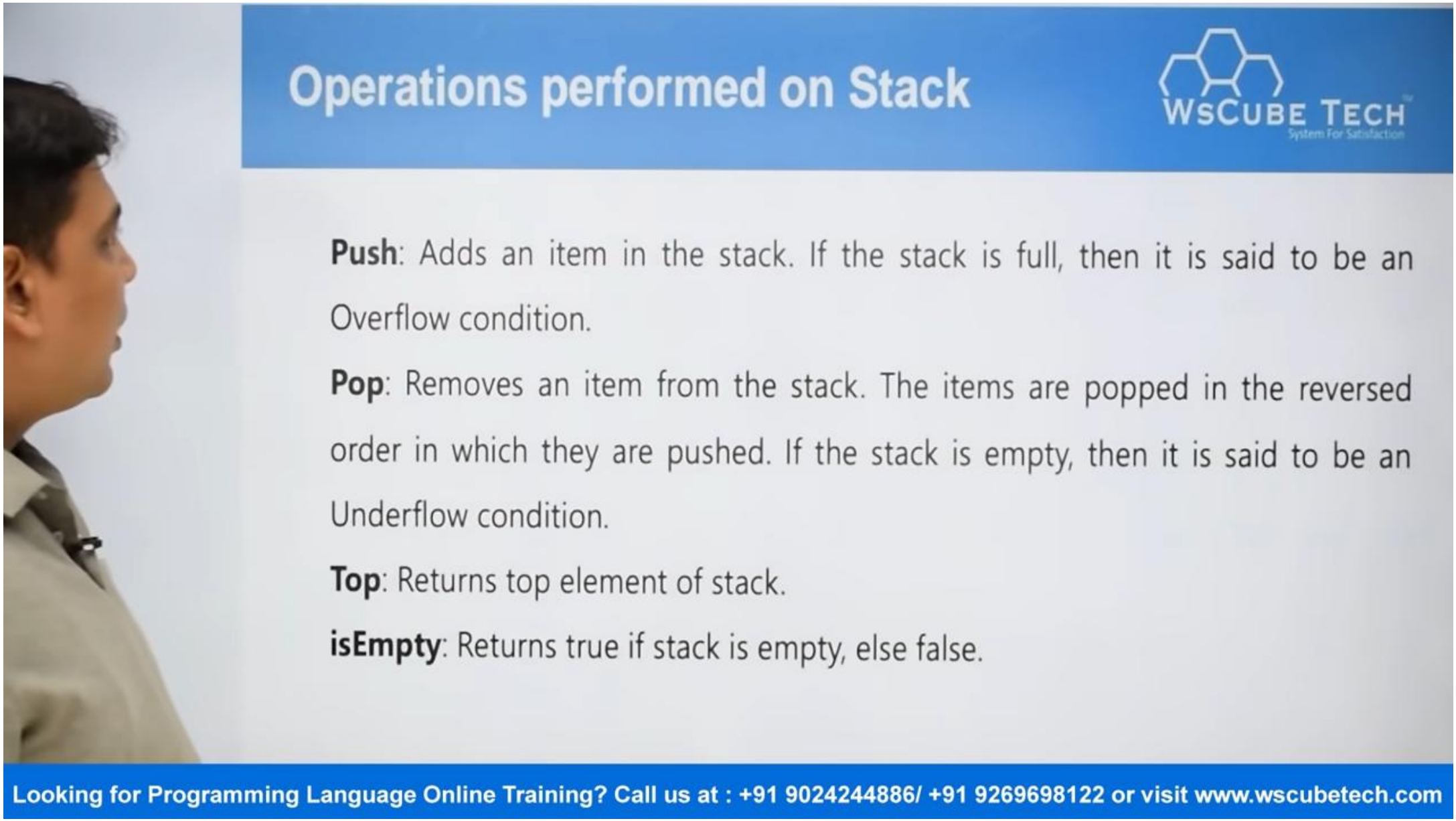
Data Structure	Time Complexity							
	Average				Worst			
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)	O(1)	O(n)	O(n)	O(n)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)

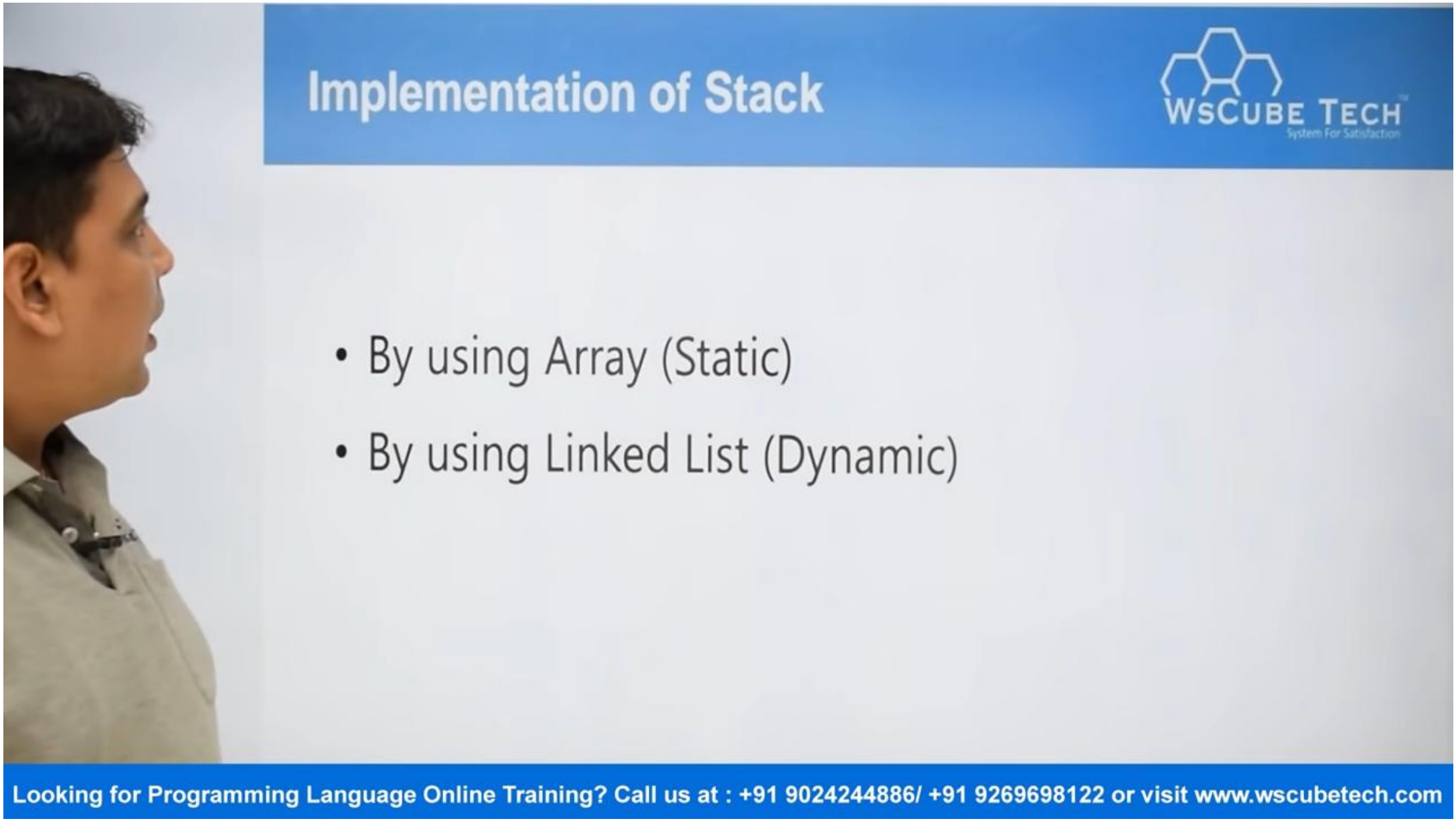
Data Structure & Algorithm

Concept of Stack

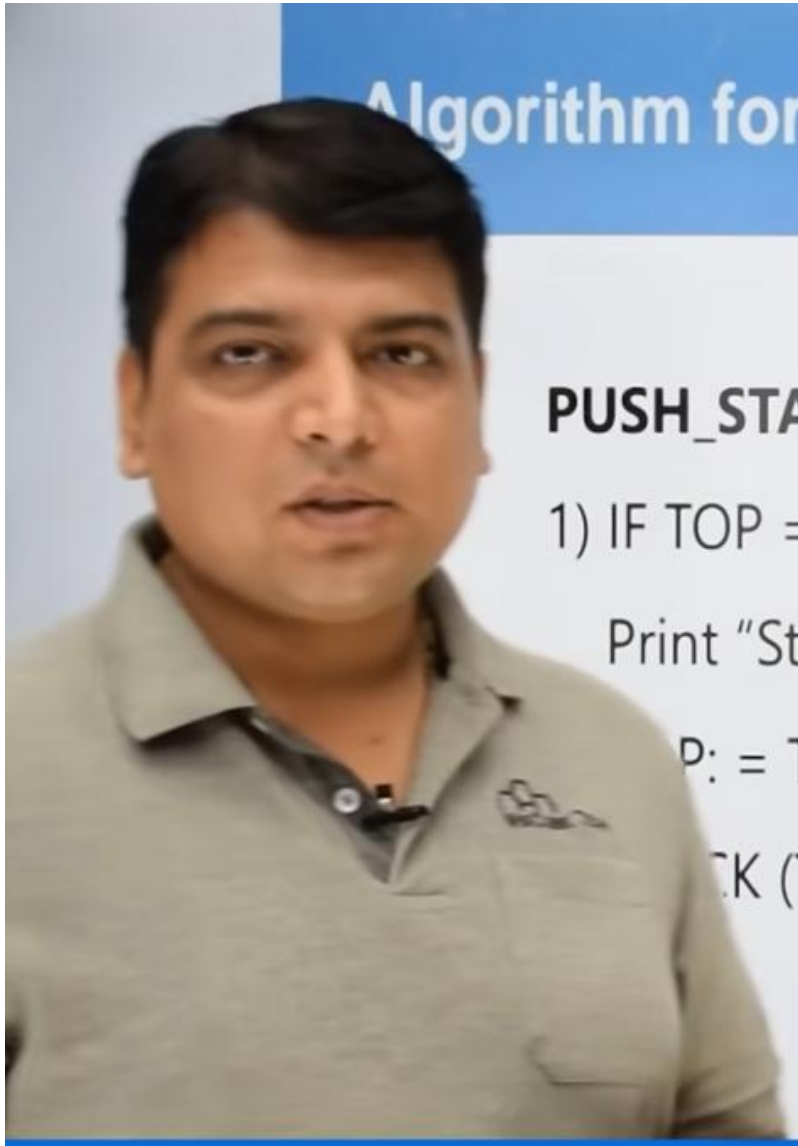

www.wscubetech.com
info@wscubetech.com
Stack in Data Structure
1st Floor, Laxmi Tower, Bhaskar Circle, Ratanada, Jodhpur.
BE TECH™
System For Satisfaction
Development – Training - Placement - Outsourcing
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com







Algorithm for Push Operation



WsCube Tech
System For Satisfaction

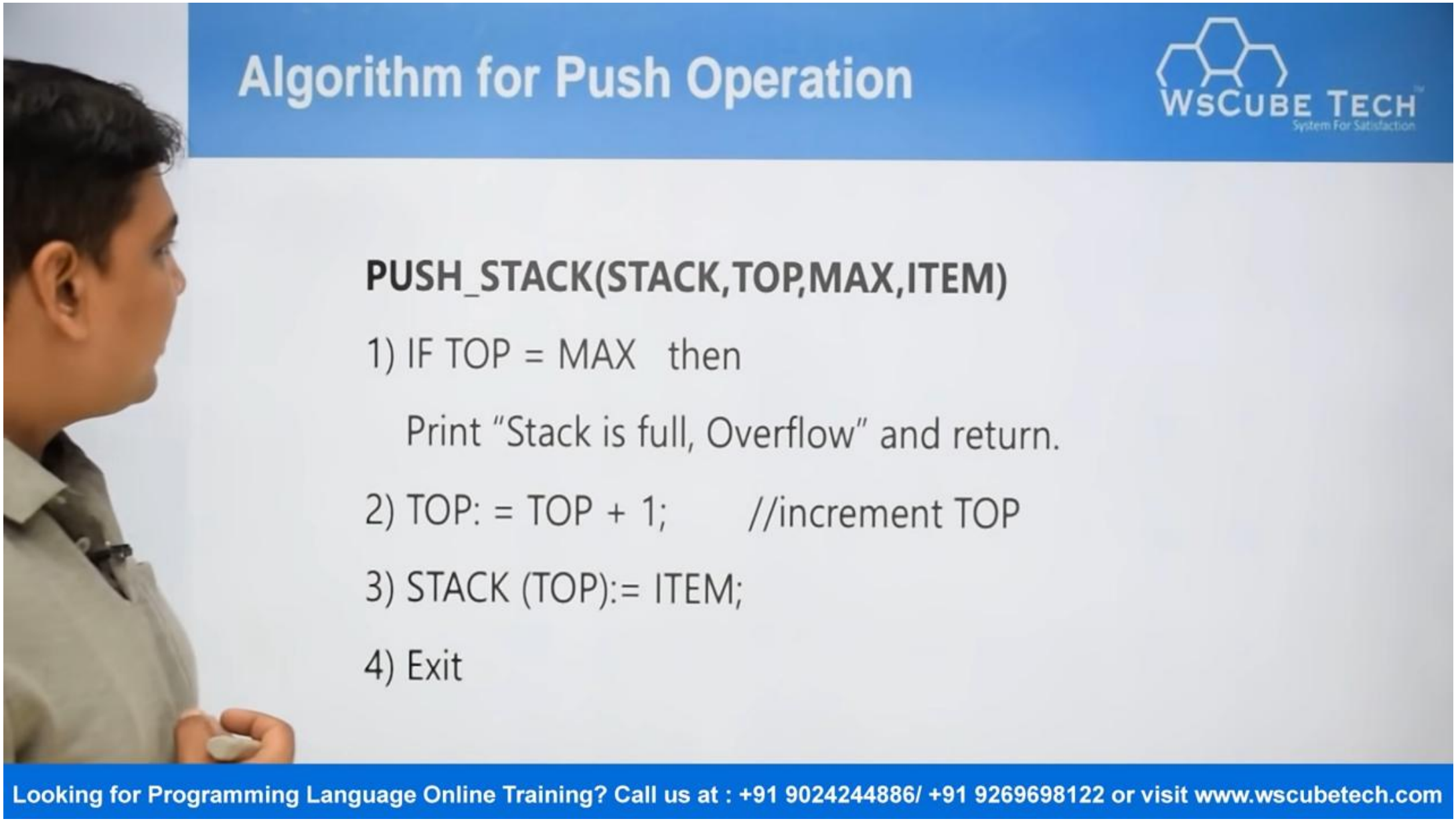
PUSH_STACK(STACK,TOP,MAX,ITEM)

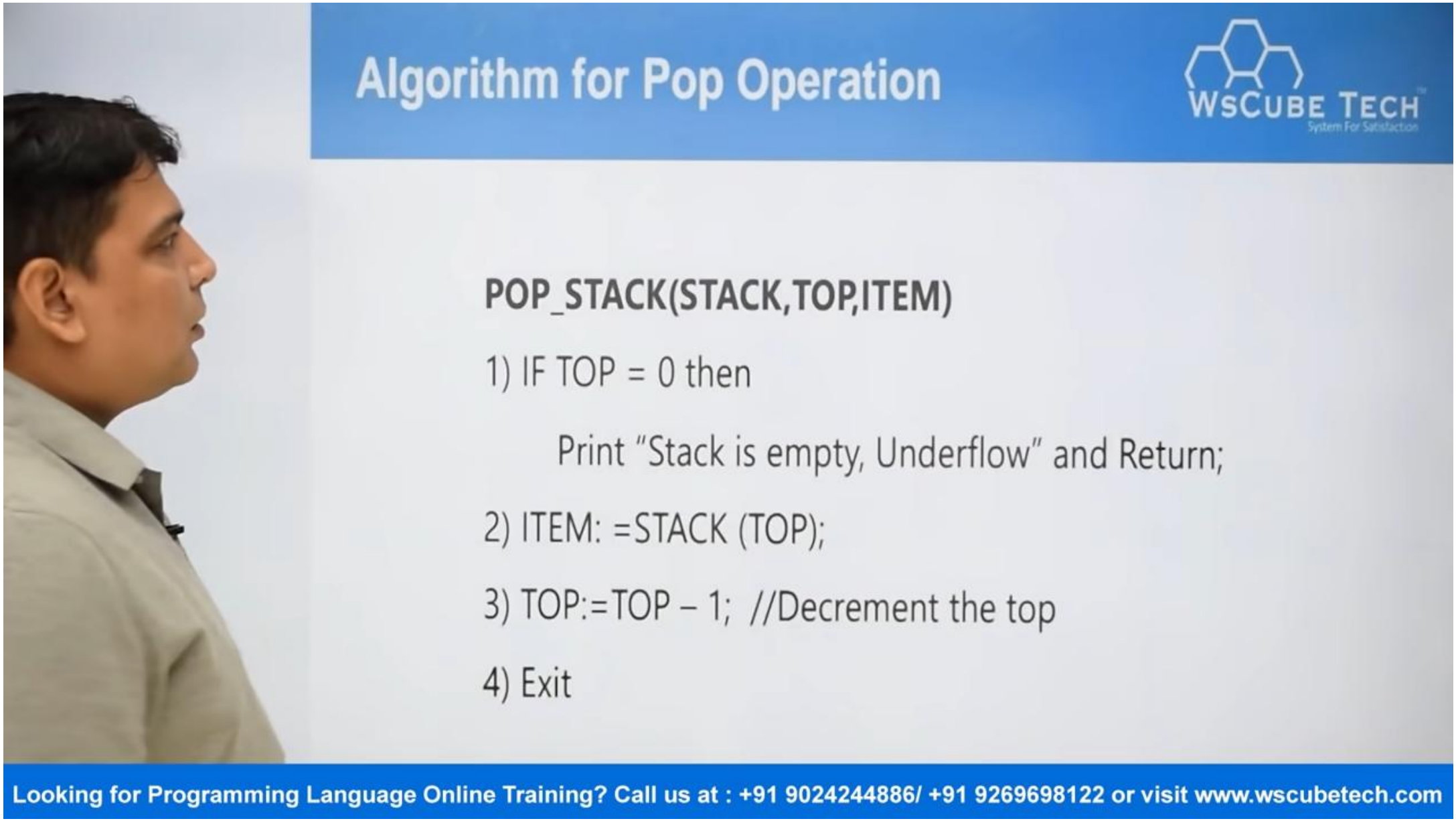
1) IF TOP = MAX then

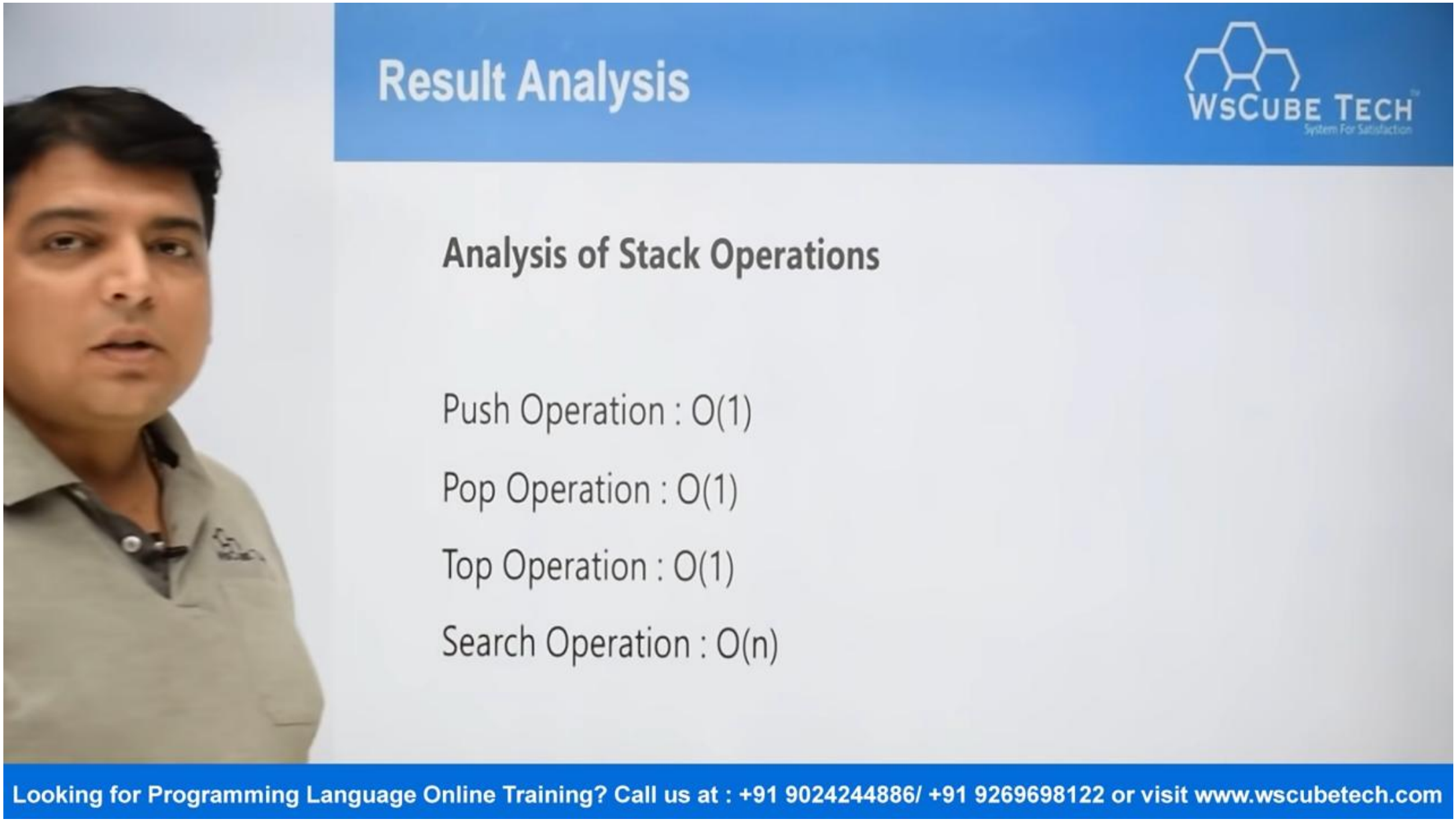
Print "Stack is full, Overflow" and return.

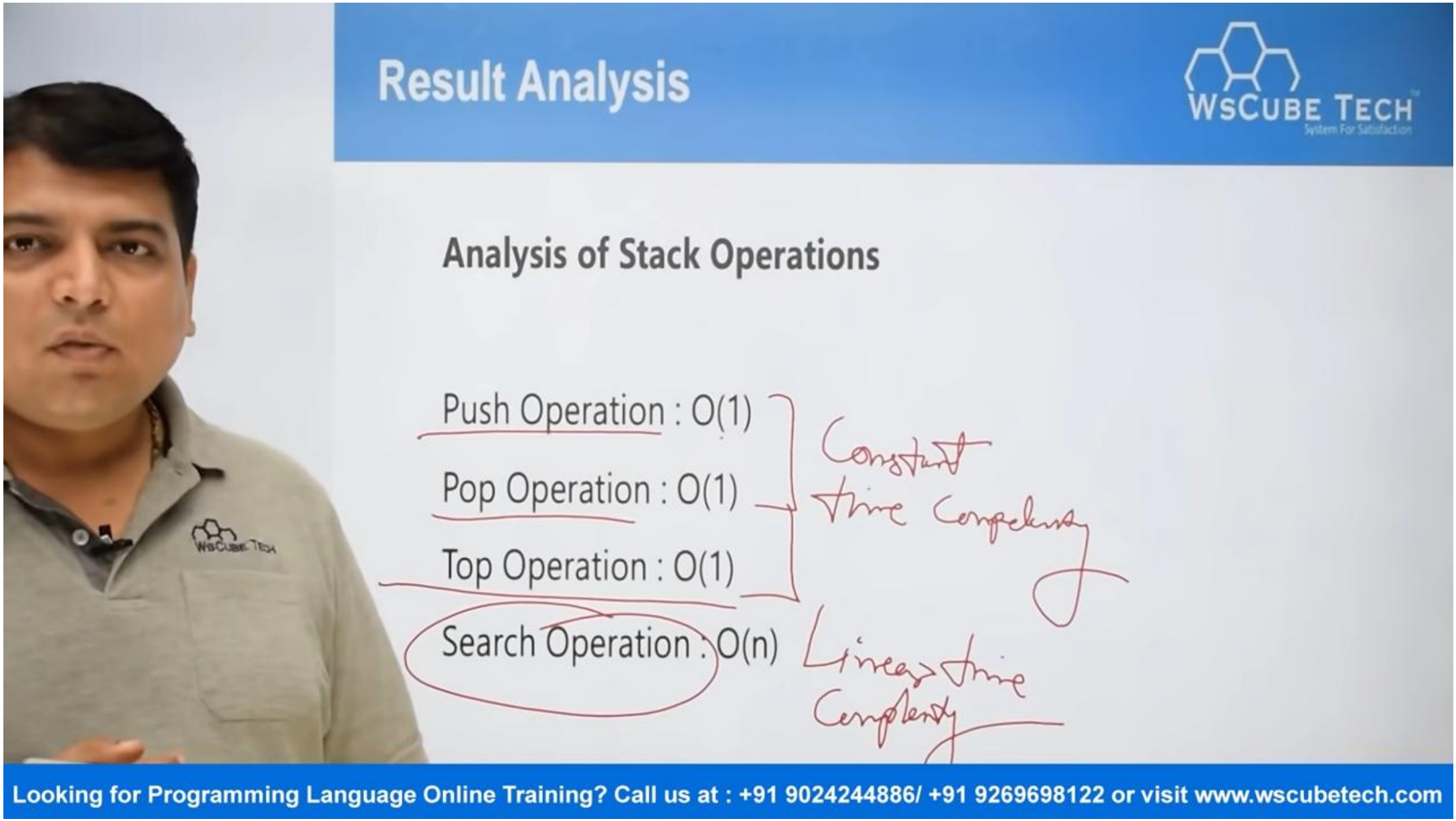
P: = TOP + 1; //increment TOP

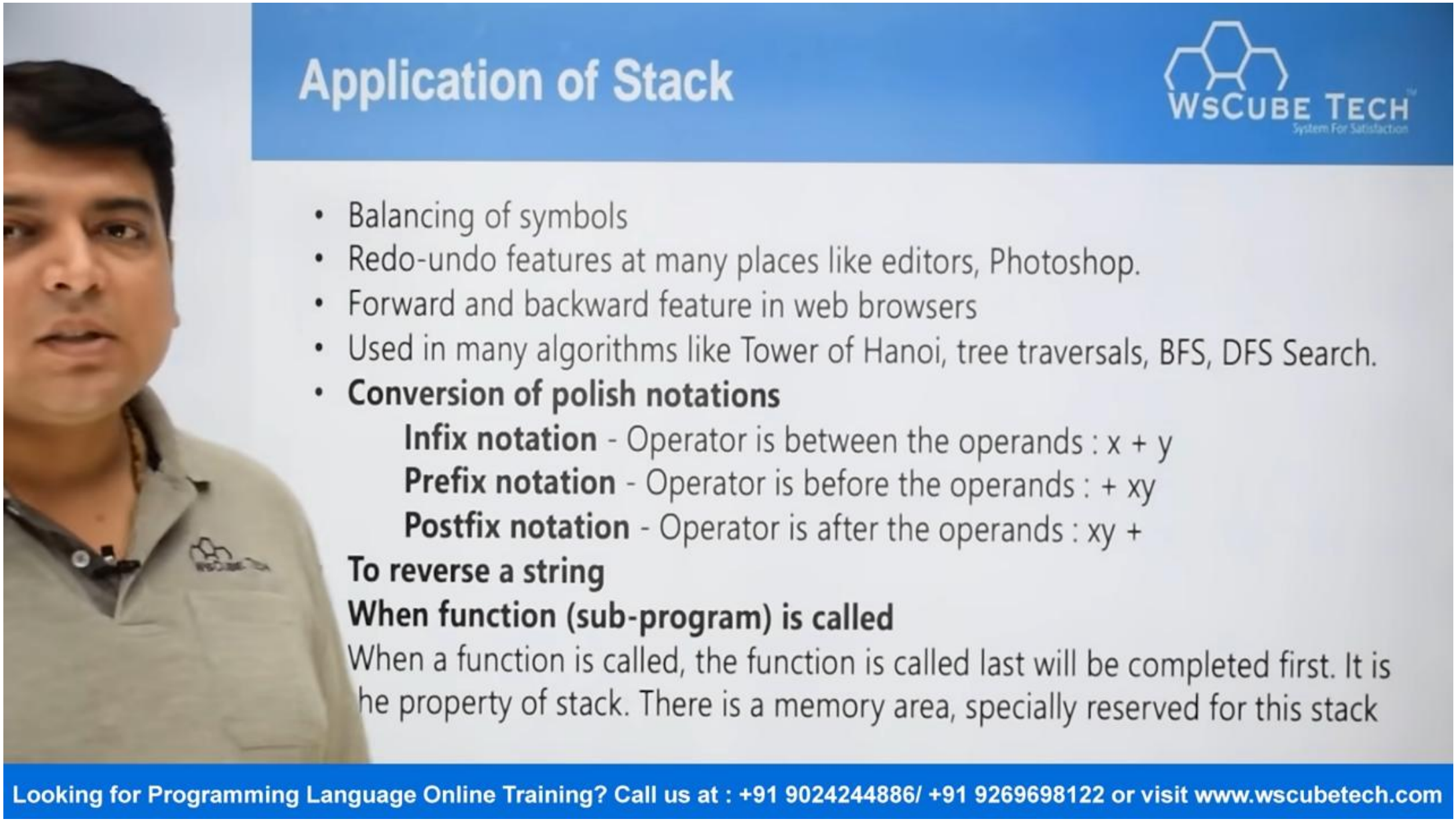
K (TOP):= ITEM;

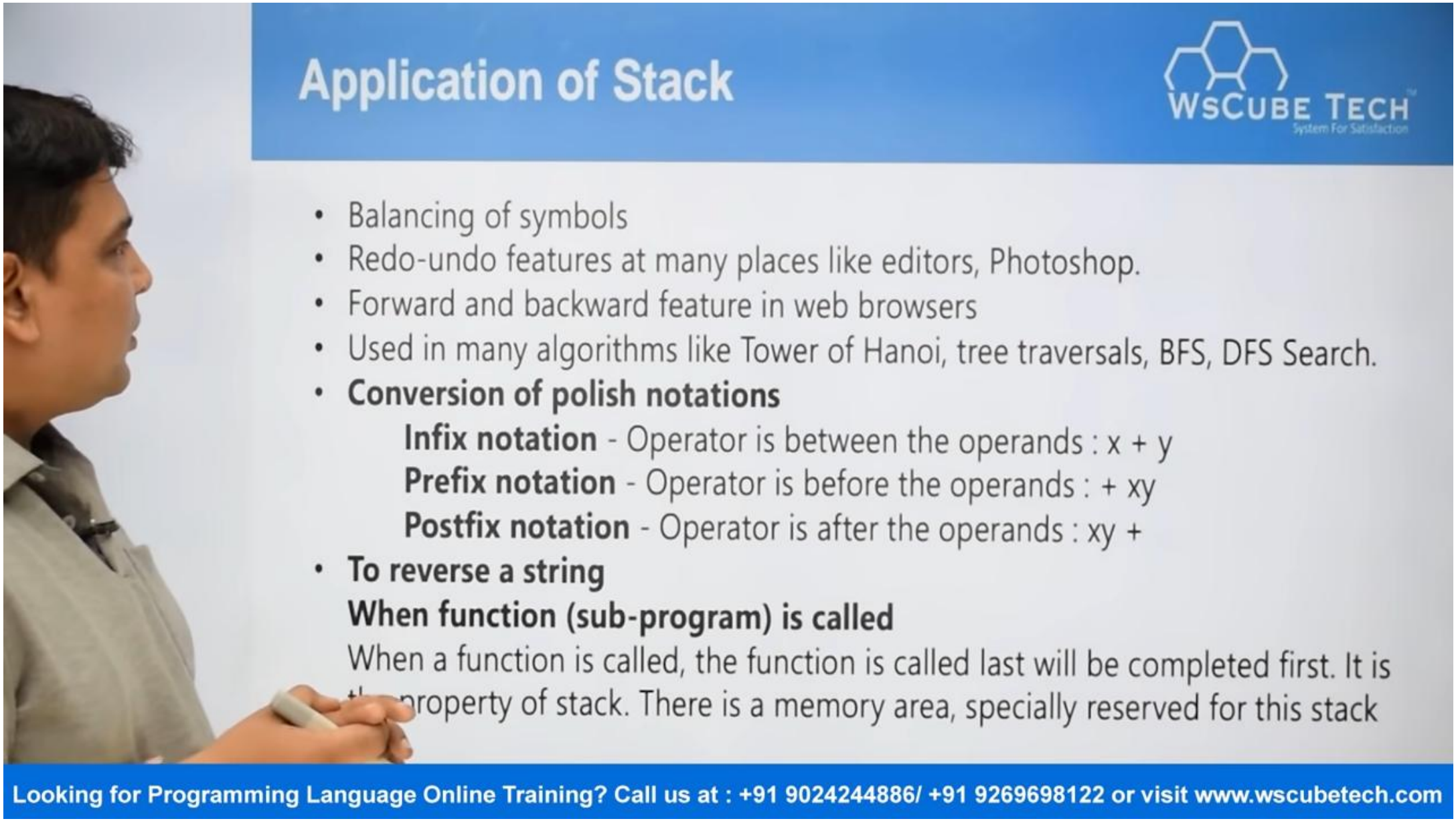








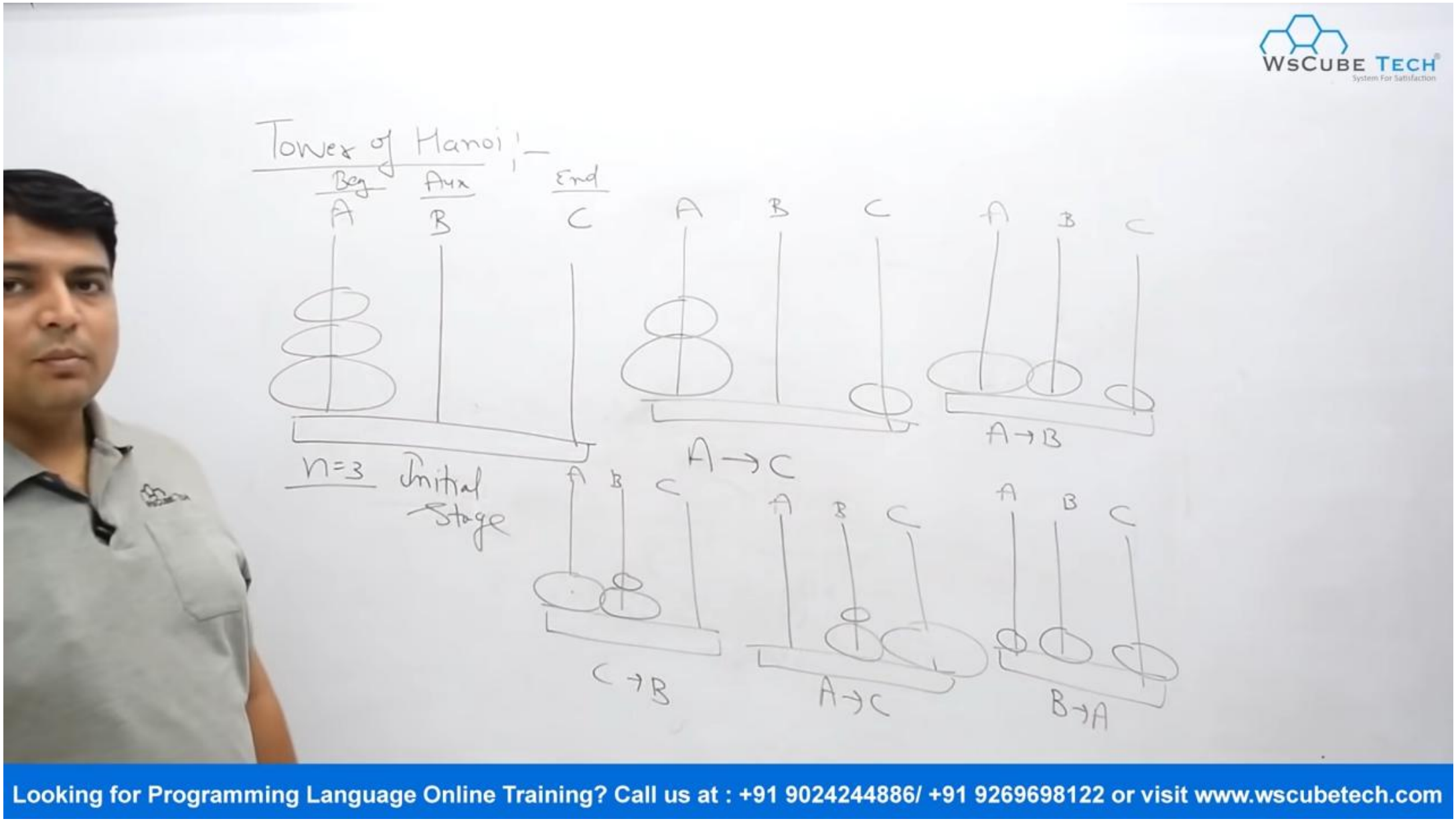




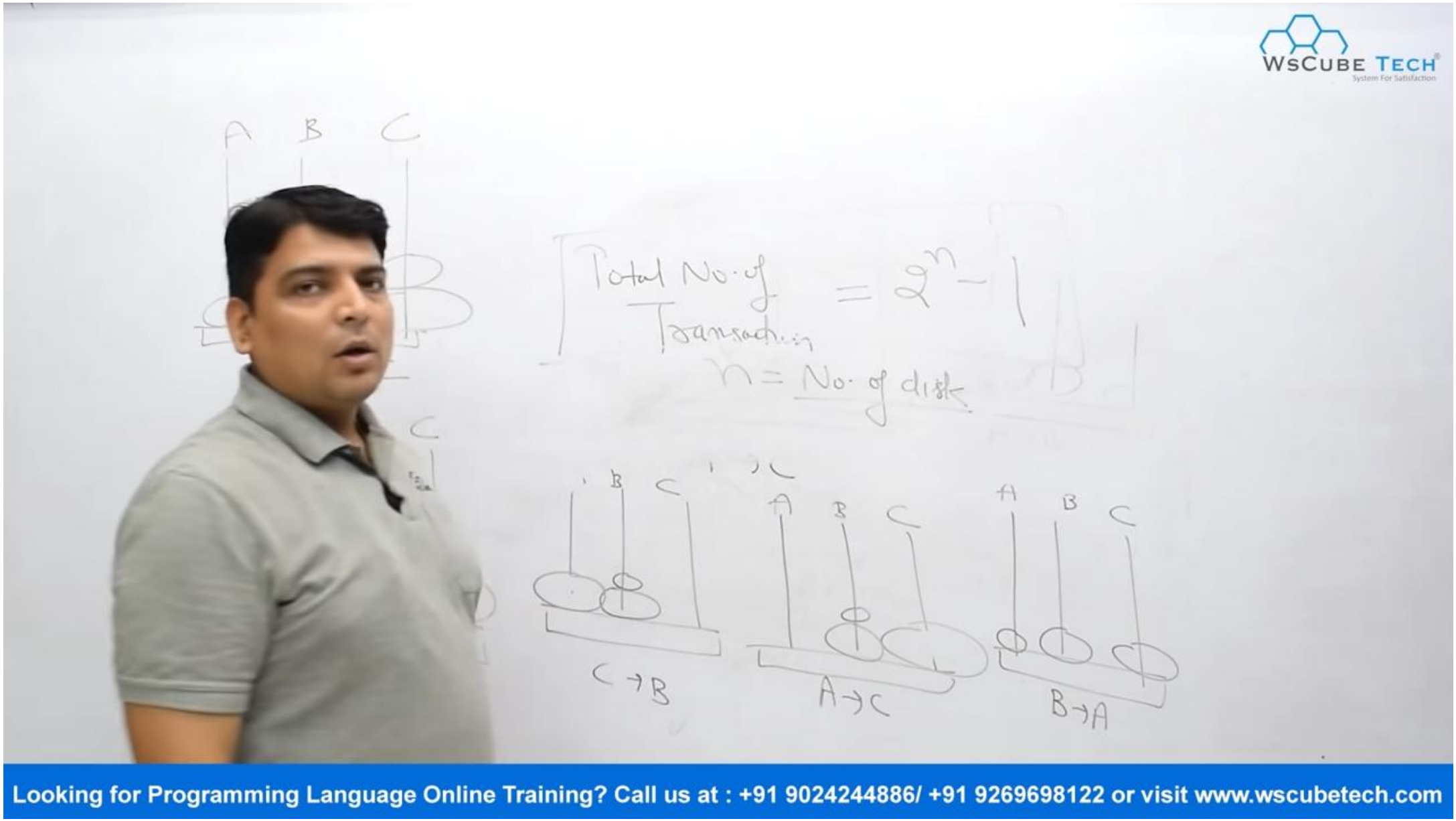
Data Structure & Algorithm

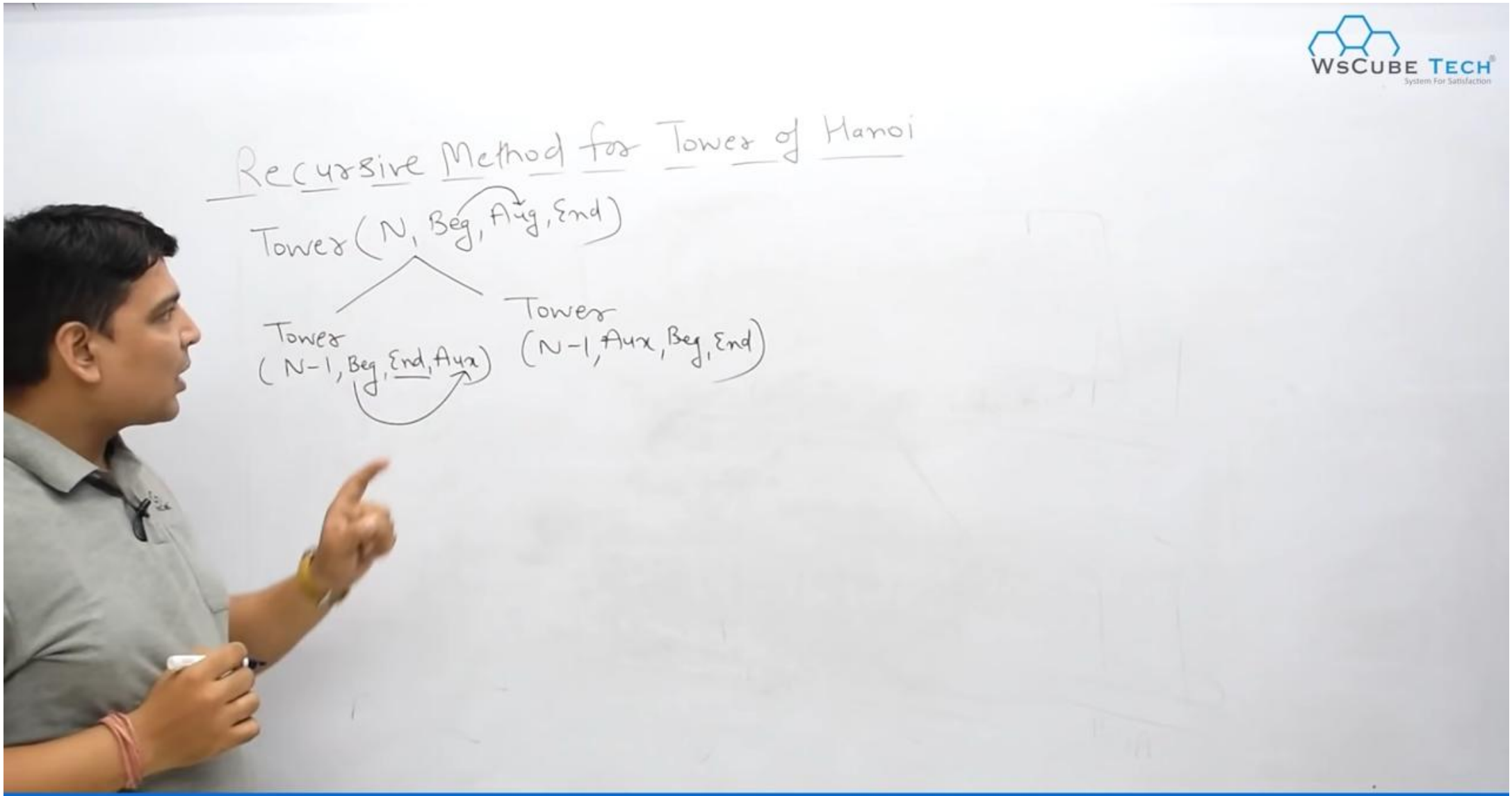
Tower of Hanoi

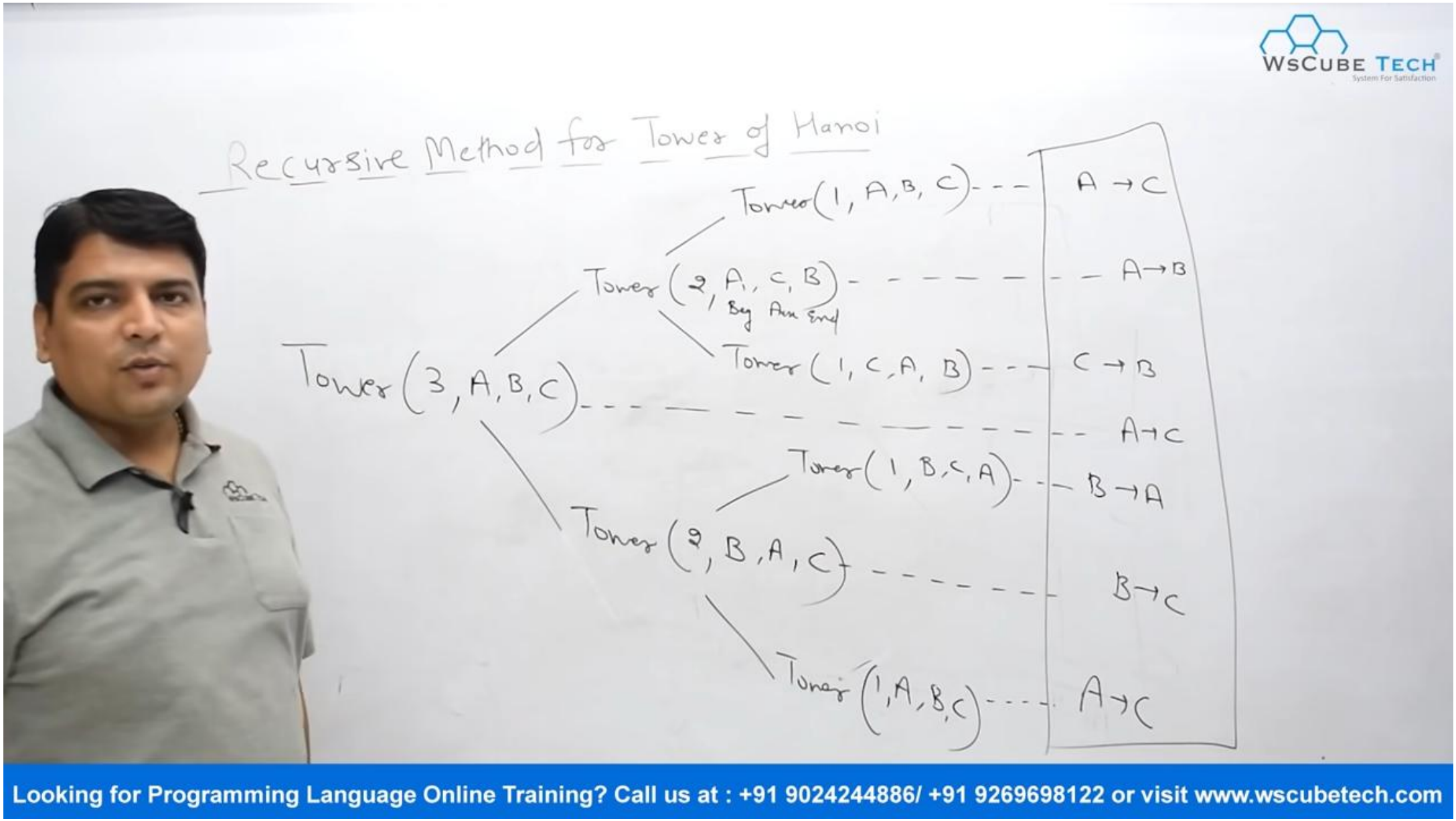
WsCube Tech™
System For Satisfaction




WsCube Tech
System For Satisfaction
A B C
B → C
A B C
A → B
A B C
A → C
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

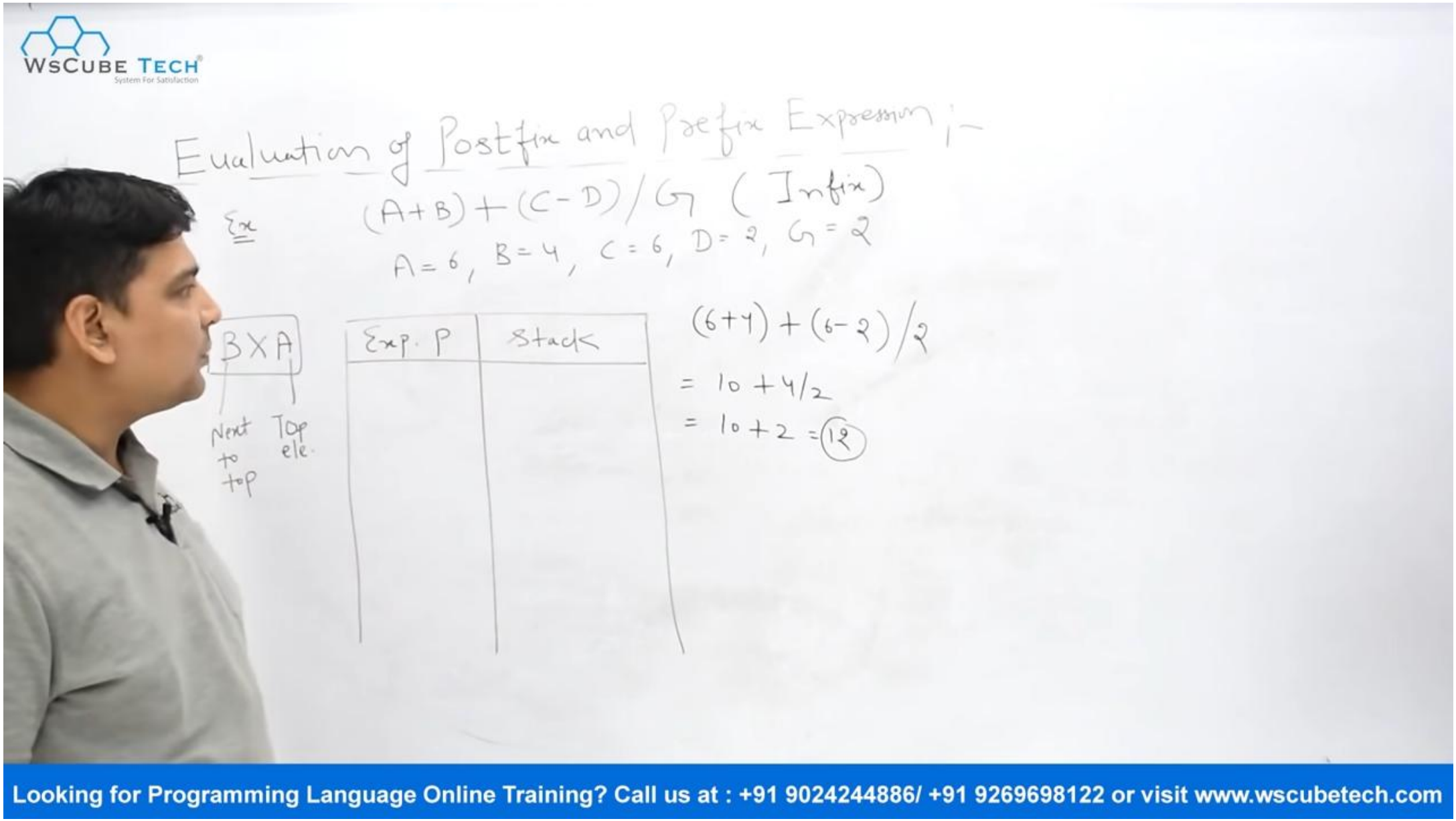

WsCube Tech
System For Satisfaction
A B C
Total No. of Transaction = $2^n - 1$
n = No. of disk
C → B
A → C
B → A
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

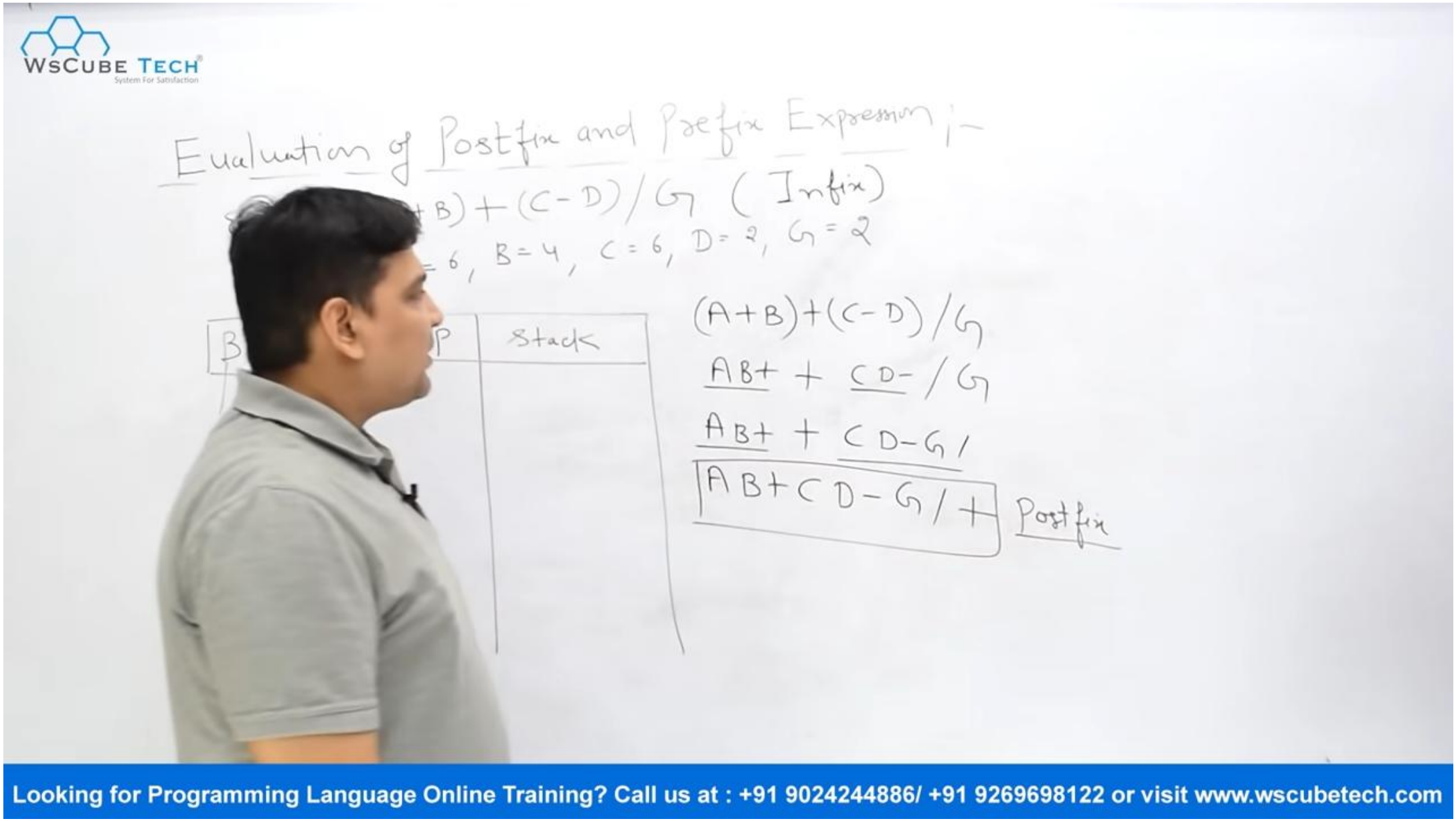


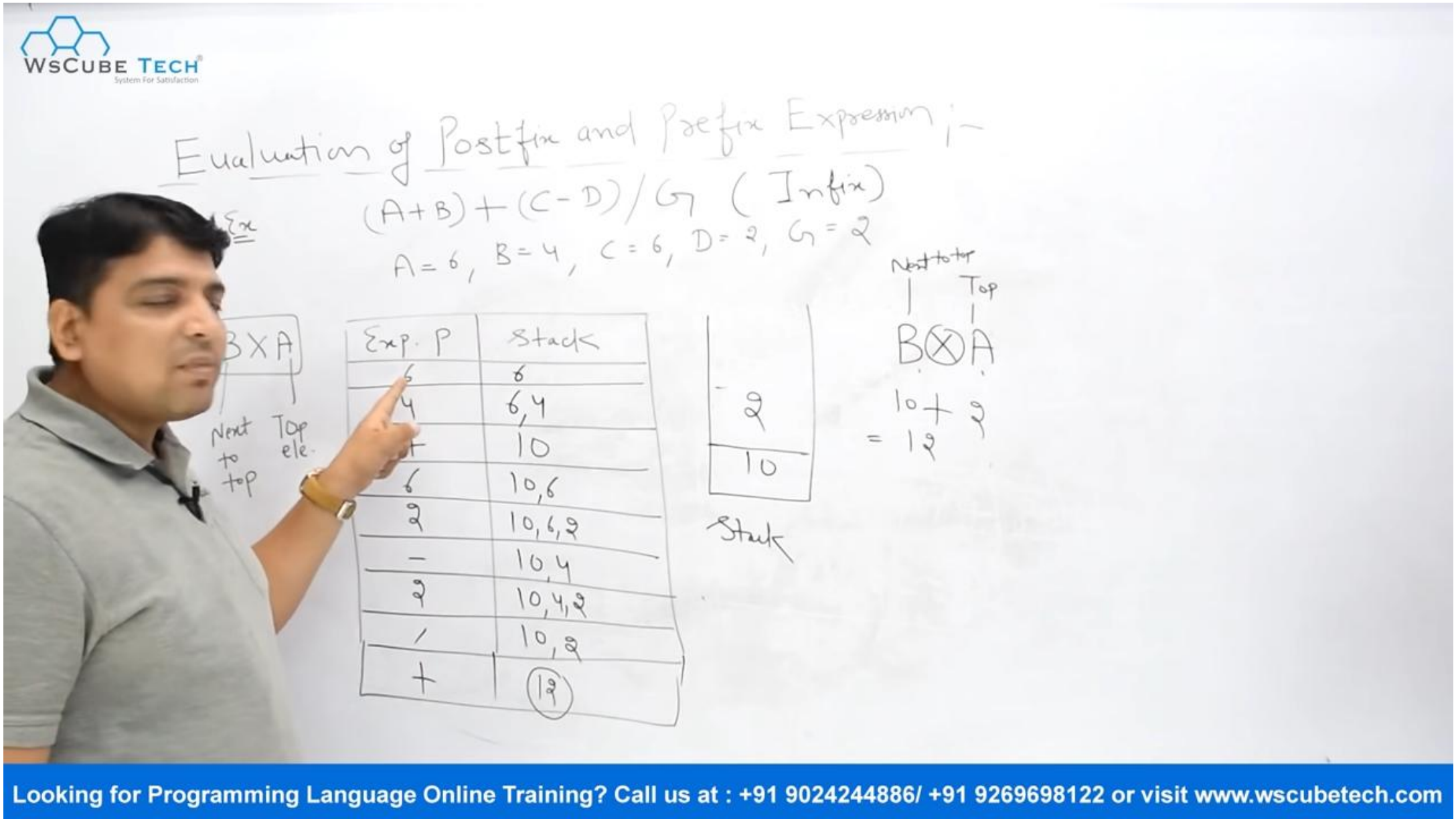


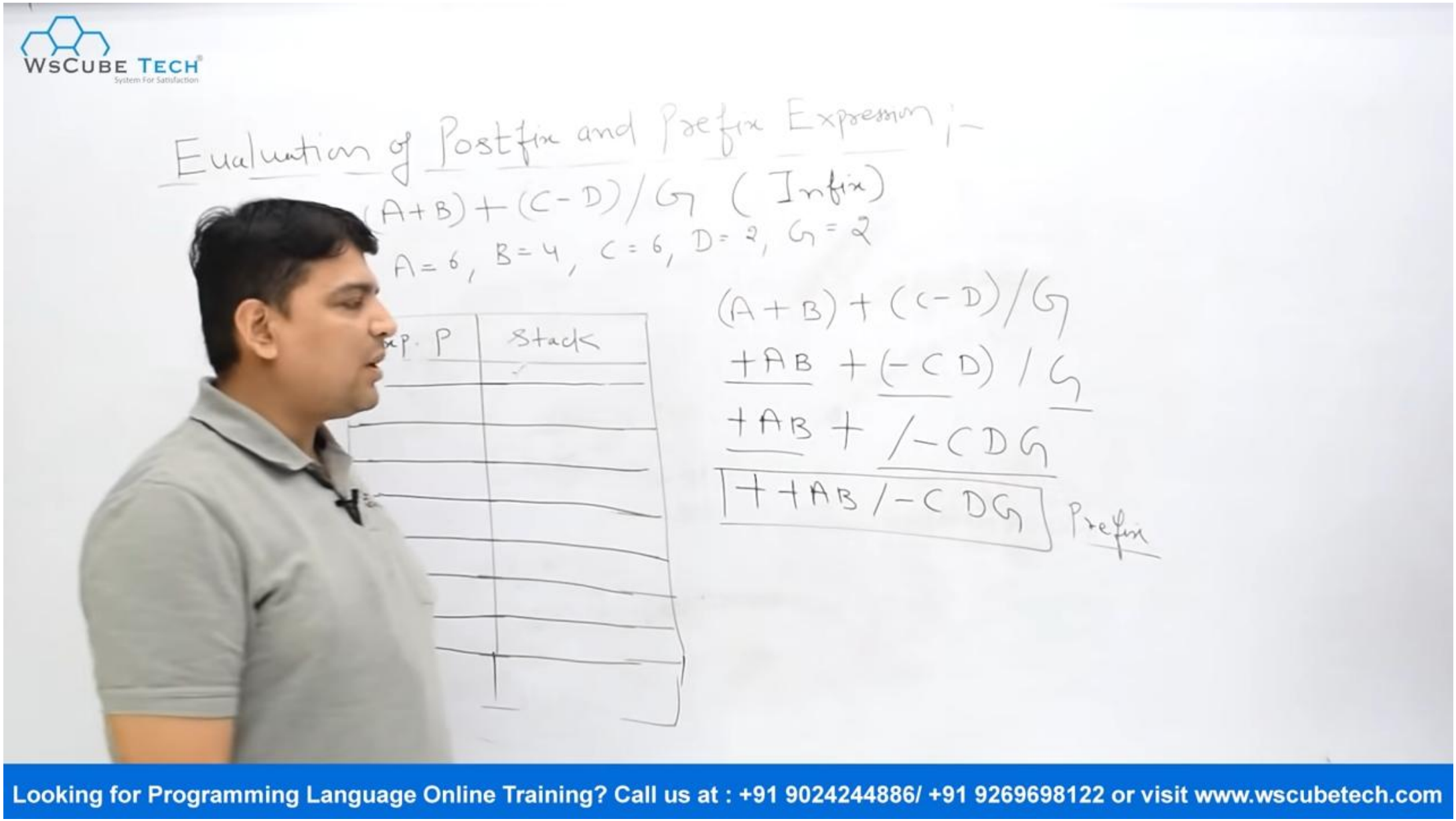
Data Structure & Algorithm

Evaluation of Postfix & Infix Expression









Evaluation of Postfix and Prefix Expression :-

(A+B)+(C-D)/G (Infix)

A=6, B=4, C=6, D=2, G=2

Ex

Exp. P	Stack
2	2
2	2,2
6	2,2,6
-	2,4
/	2
4	2,4
6	2,4,6
+	2,10
+	

top Next

B ⊗ A

6 + 4

= 10

Data Structure & Algorithm

Infix to Postfix Conversion

WsCube Tech
System For Satisfaction

Converting Infix Expression into Postfix Expression

1) ((A-B)*D) $ ((E+F)/G)
2) A+(B*C+(D/E$F)/G)/M

WsCube Tech
System For Satisfaction

Converting Infix Expression into Postfix Expression

1) ((A-B)*D)$((E+F)/G)

2) A+(B*C+(D/E$F)/G)/M

((A-B)*D)$((E+F)/G)

((AB-)*D)$((EF+)/G)

(AB-D*)$(EF+G/)

AB-D*EF+G/$ Postfix

$ ↑	R→L
/, *	L→R
+, -	L→R

WsCube Tech
System For Satisfaction

Converting Infix Expression into Postfix Expression

1) ((A-B)*D)$((E+F)/G)

2) A+(B*C+(D/E$F)/G)/M

((A-B)*D)$((E+F)/G)

((-AB)*D)$((+EF)/G)

*-ABD$/+EFG

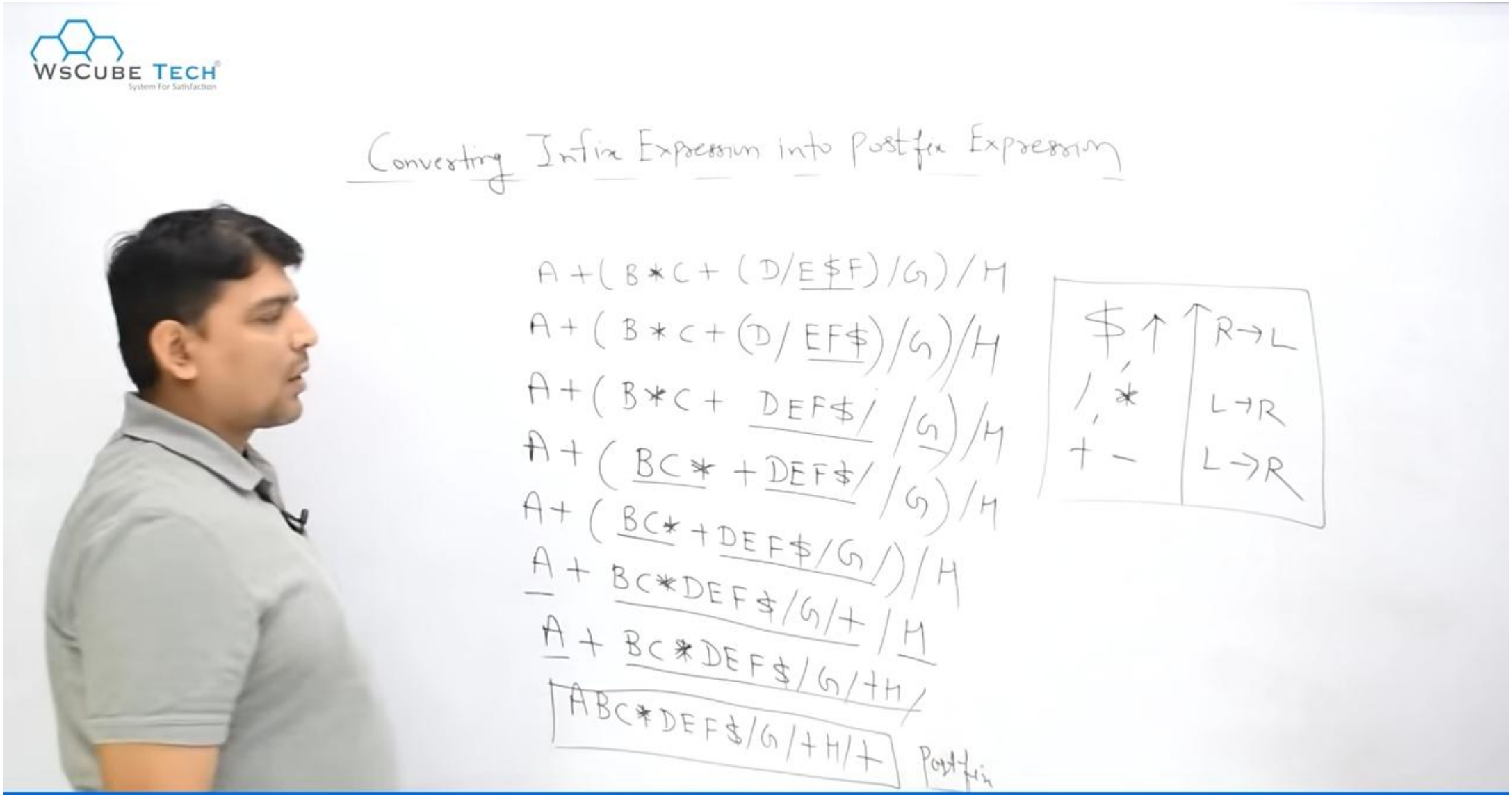
$*-ABD/+EFG

Prefix Exp

$ ↑	R→L
/, *	L→R
+, -	L→R

WsCube Tech
System For Satisfaction

Converting Infix Expression into Postfix Expression



Data Structure & Algorithm

Infix to PostfixExpression

WsCube Tech
System For Satisfaction

Converting Infix Expression into Postfix Expression (Stack Concept)

Ex: (A - B$D) / E+F -G

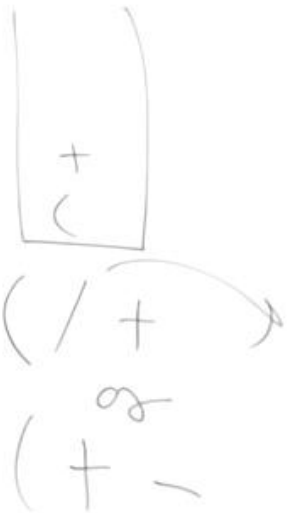
Exp S	Stack	Exp P
	(	
(	((	

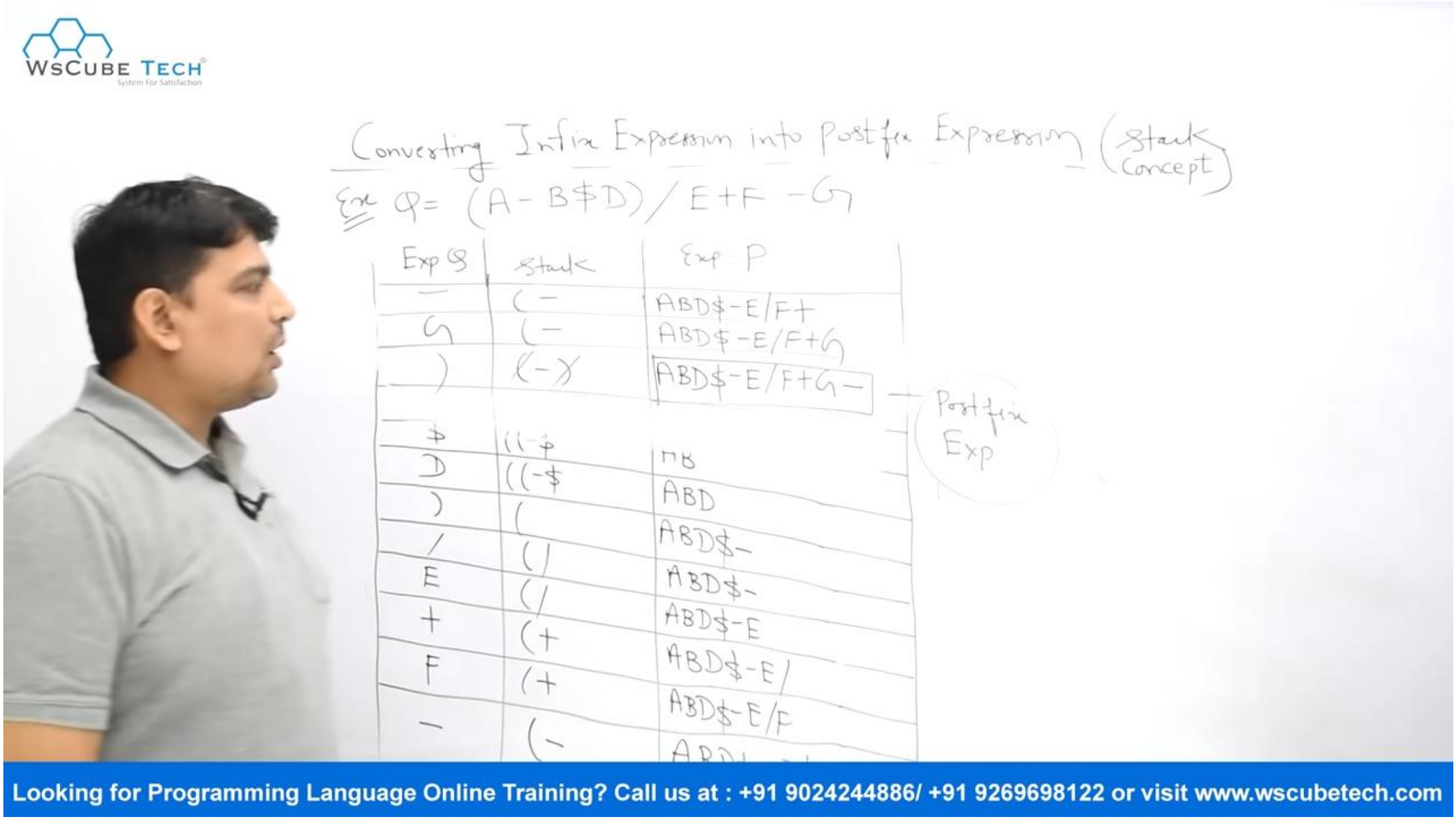
WsCube Tech
System For Satisfaction

Converting Infix Expression into Postfix Expression (Stack Concept)

Ex Q = (A - B$D) / E + F - G

Exp Q	stack	Exp P
	(	
(	((	
A	((	A
-	((-	A
B	((-	AB
$	((-$	AB
D	((-$	ABD
)	(	ABD$-
/	(/	ABD$-
E	(/	ABD$-E
+	(+	ABD$-E/
F	(+	ABD$-E/F
-	(-	





Data Structure & Algorithm

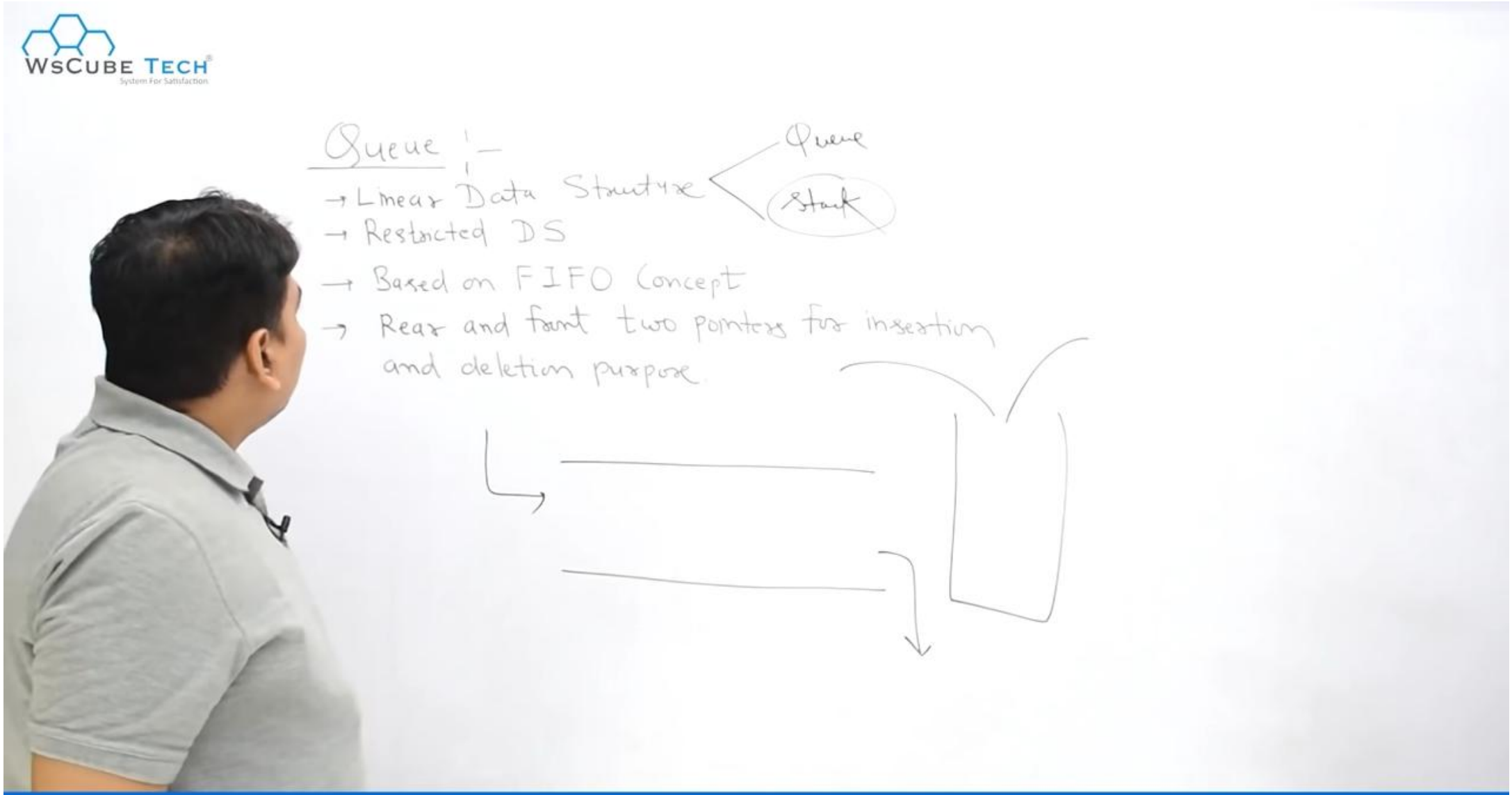
Queue in DSA

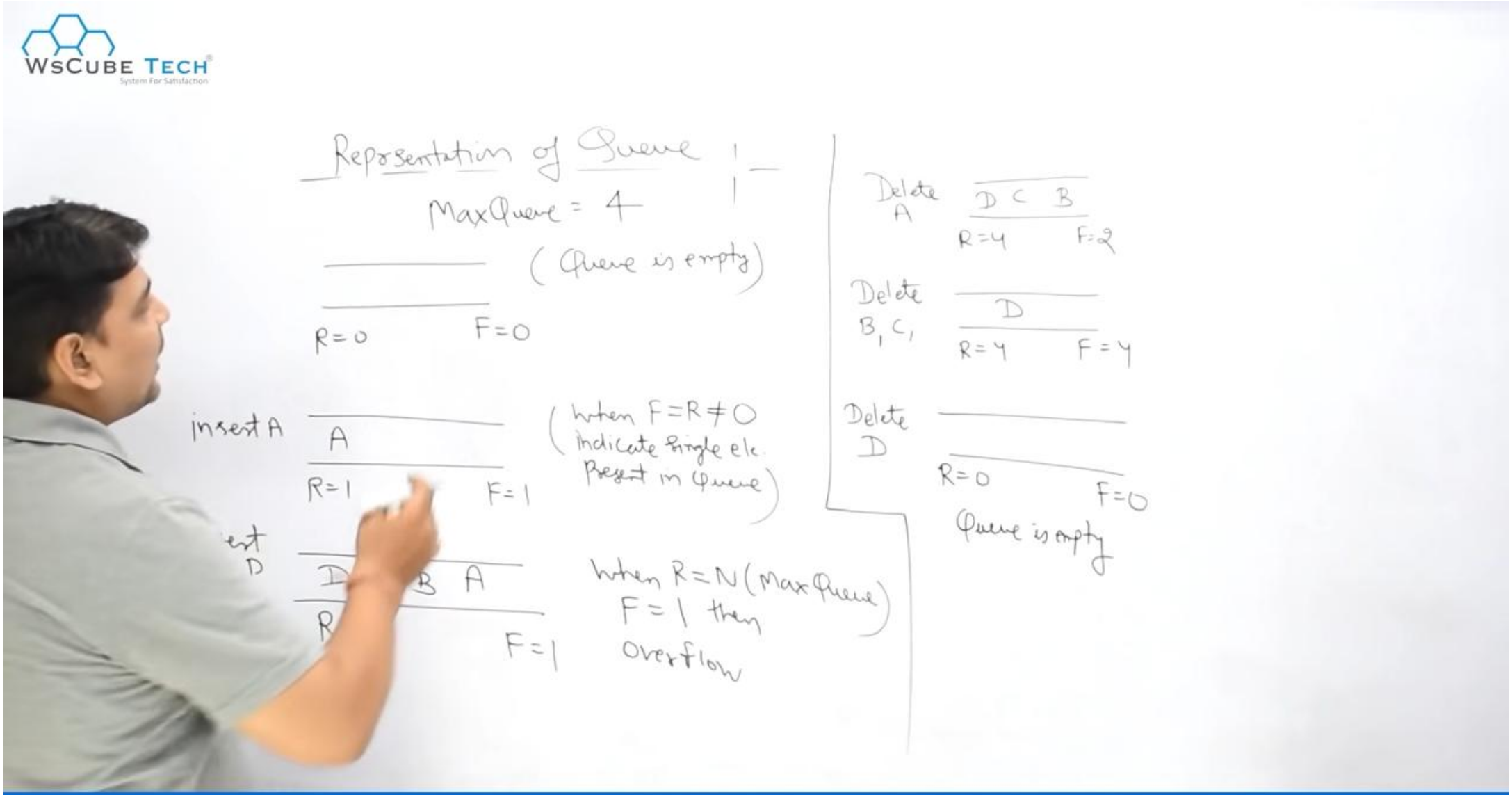
WsCube Tech
System For Satisfaction

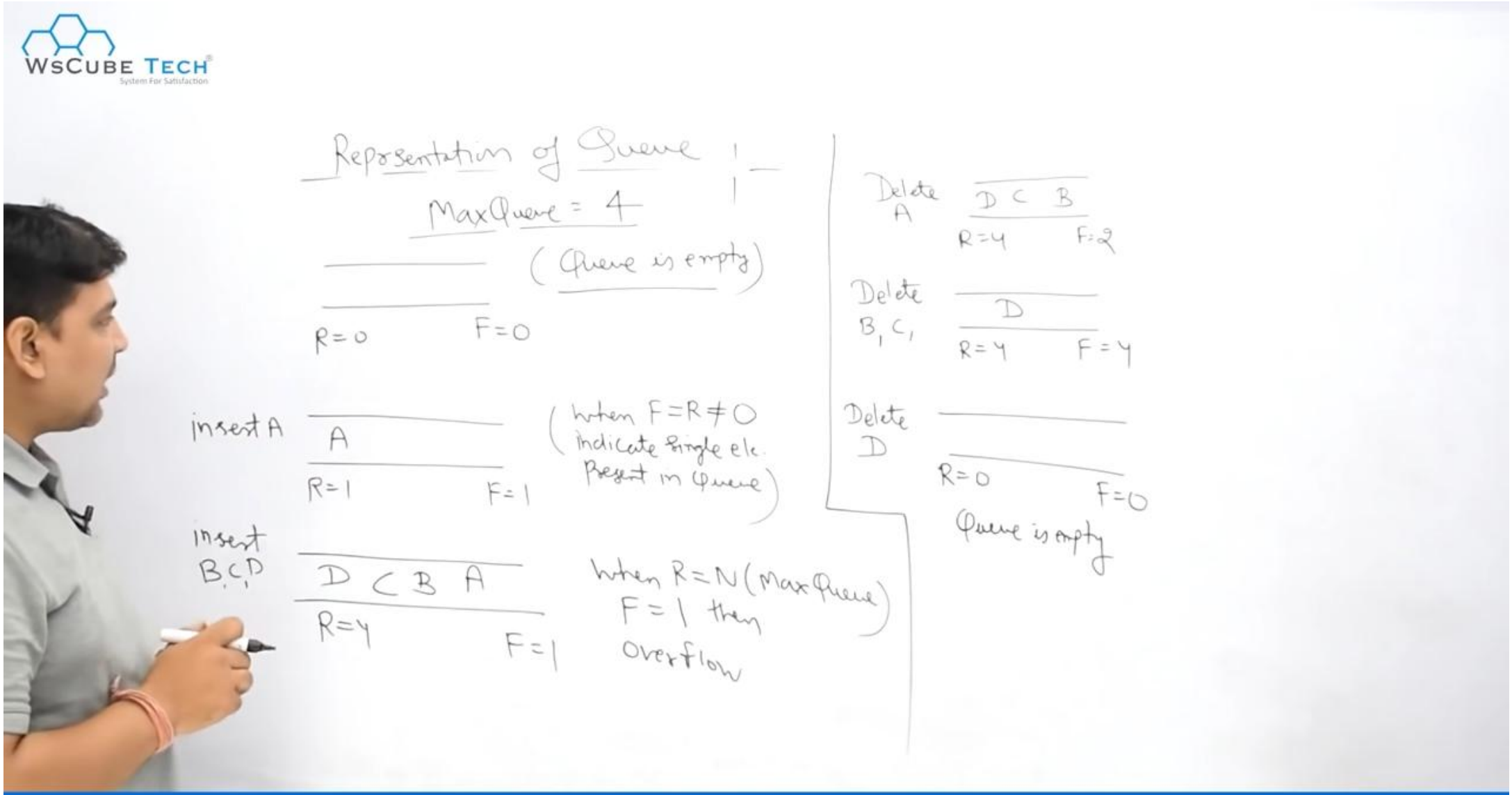
Queue :-

- → Linear Data Structure
- → Restricted DS
- → Based on FIFO Concept
- → Rear and front two pointers for insertion and deletion purpose

WsCube Tech
System For Satisfaction

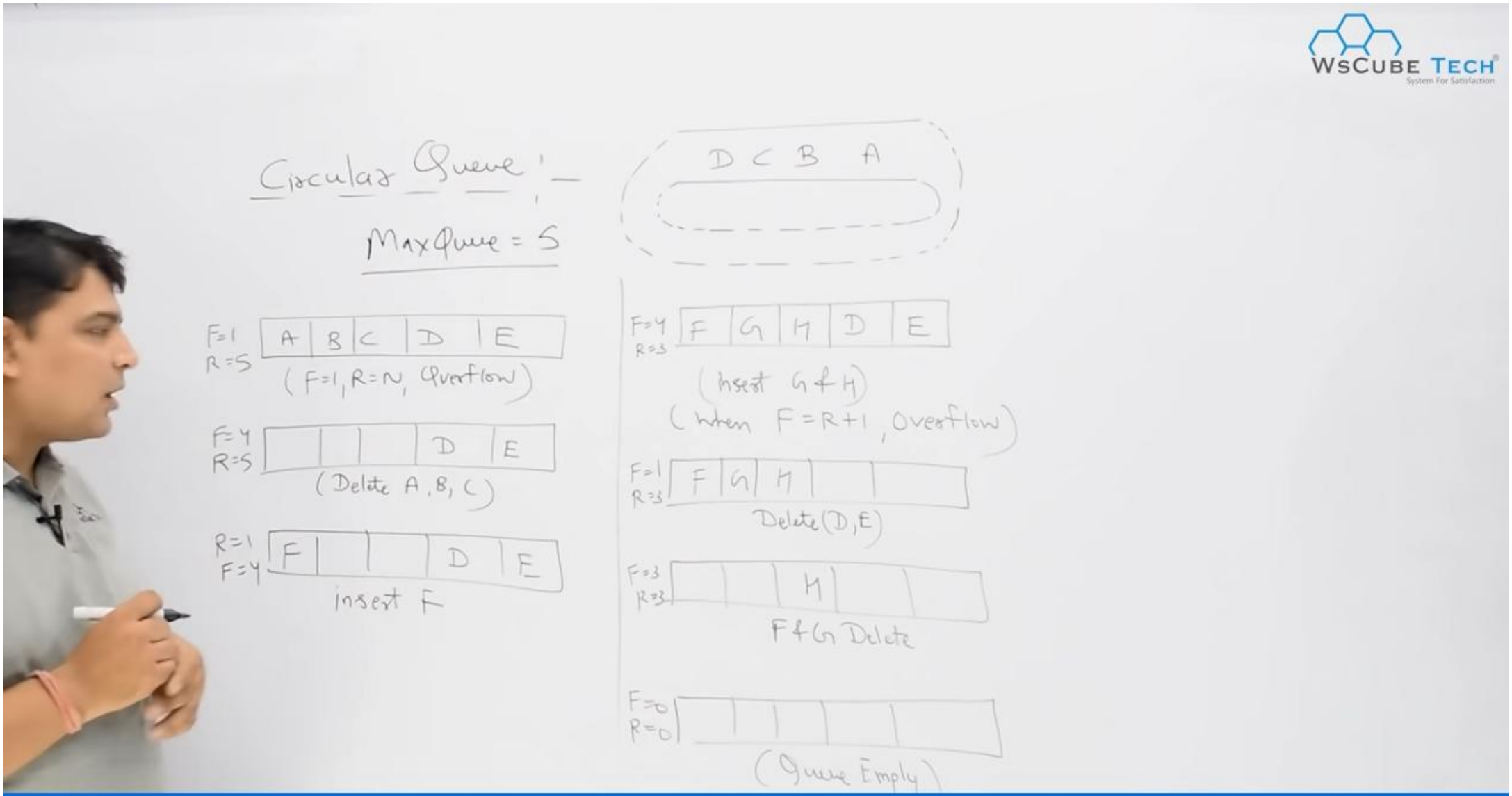






Data Structure & Algorithm

Circular Queue



Circular Queue Algorithm :-

Insertion

CQI(Queue, N, F, R, item)

1) if F=1, R=N or
   F=R+1, Overflow & Return
2) if F=Null
   Set F=1, R=1
   else if R=N then
   Set R=1
   else R=R+1
3) Set Queue[R]=item
4) Return

Deletion

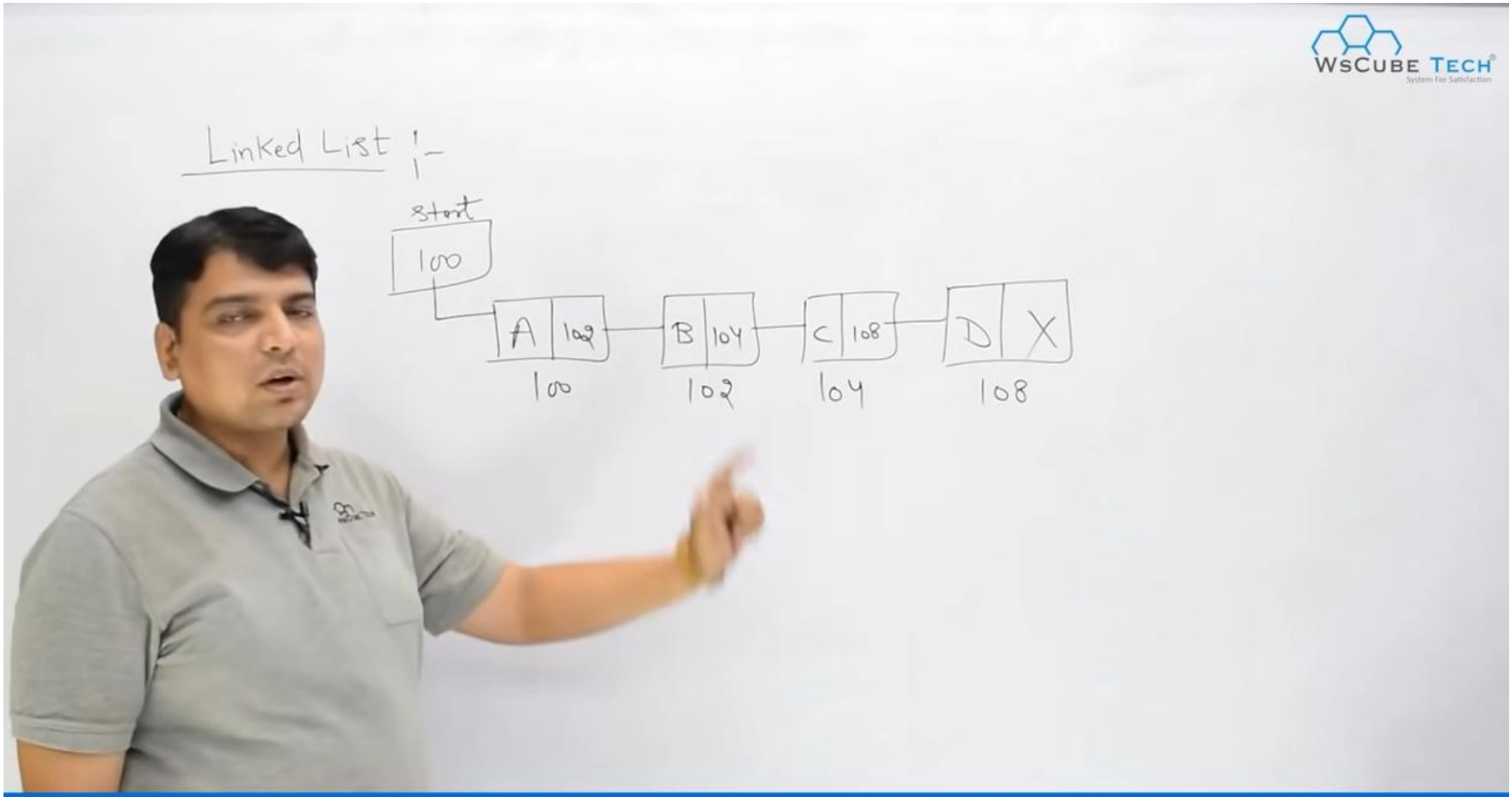
CQD(Queue, N, F, R, item)

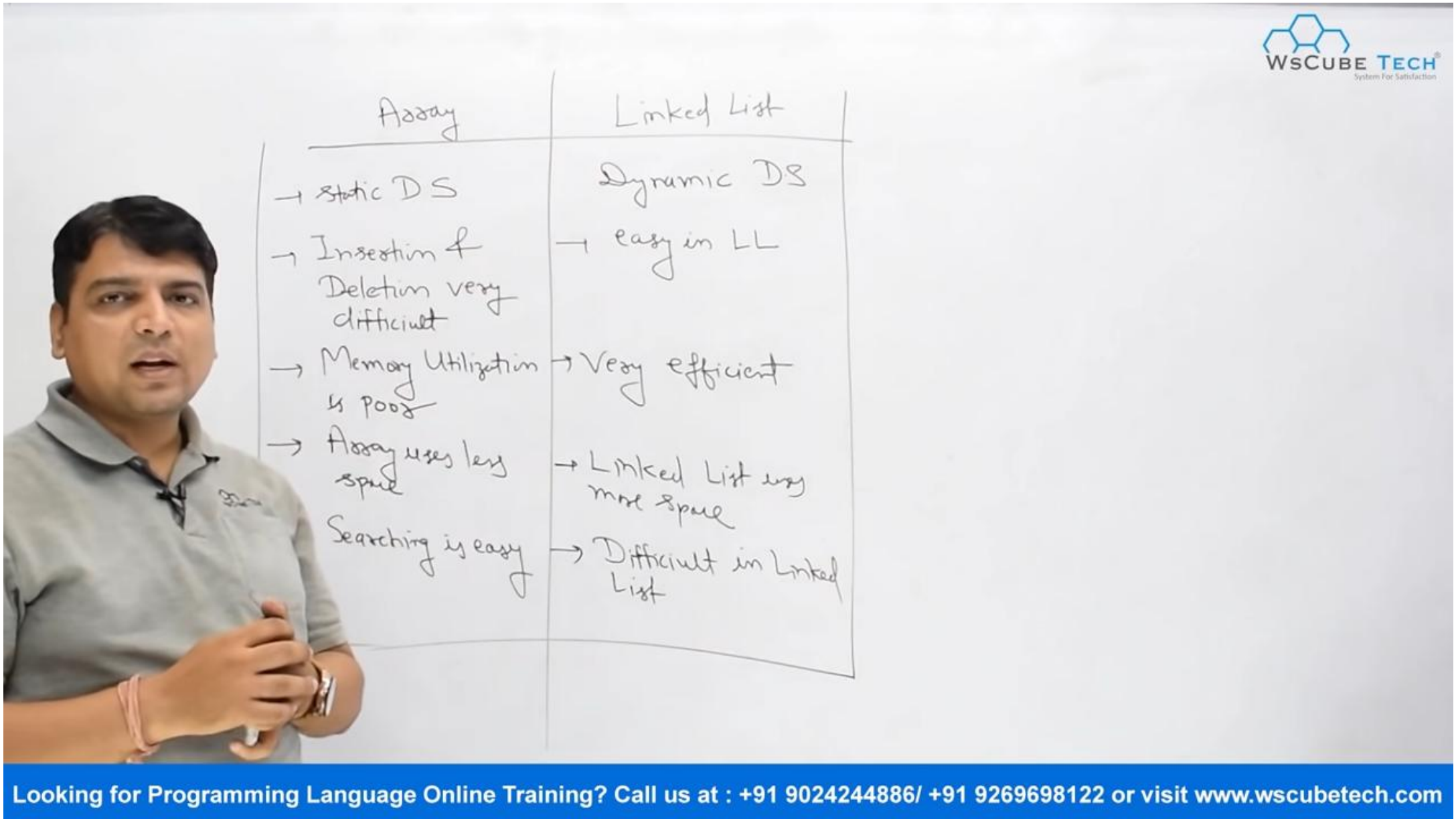
1) if F=Null, write Underflow & Return
2) set item = Queue[F]
3) if F=R≠Null
   set F=R=Null
   else if F=N set F=1
   else F=F+1
4) Return

Data Structure & Algorithm

Linked List in DSA

WsCube Tech™
System For Satisfaction

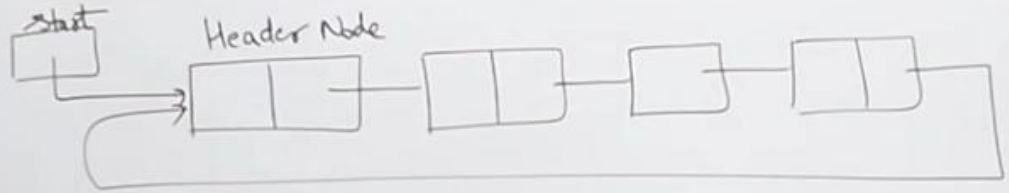




Data Structure & Algorithm

Circular Linked List

Circular Linked List :-



Advantages

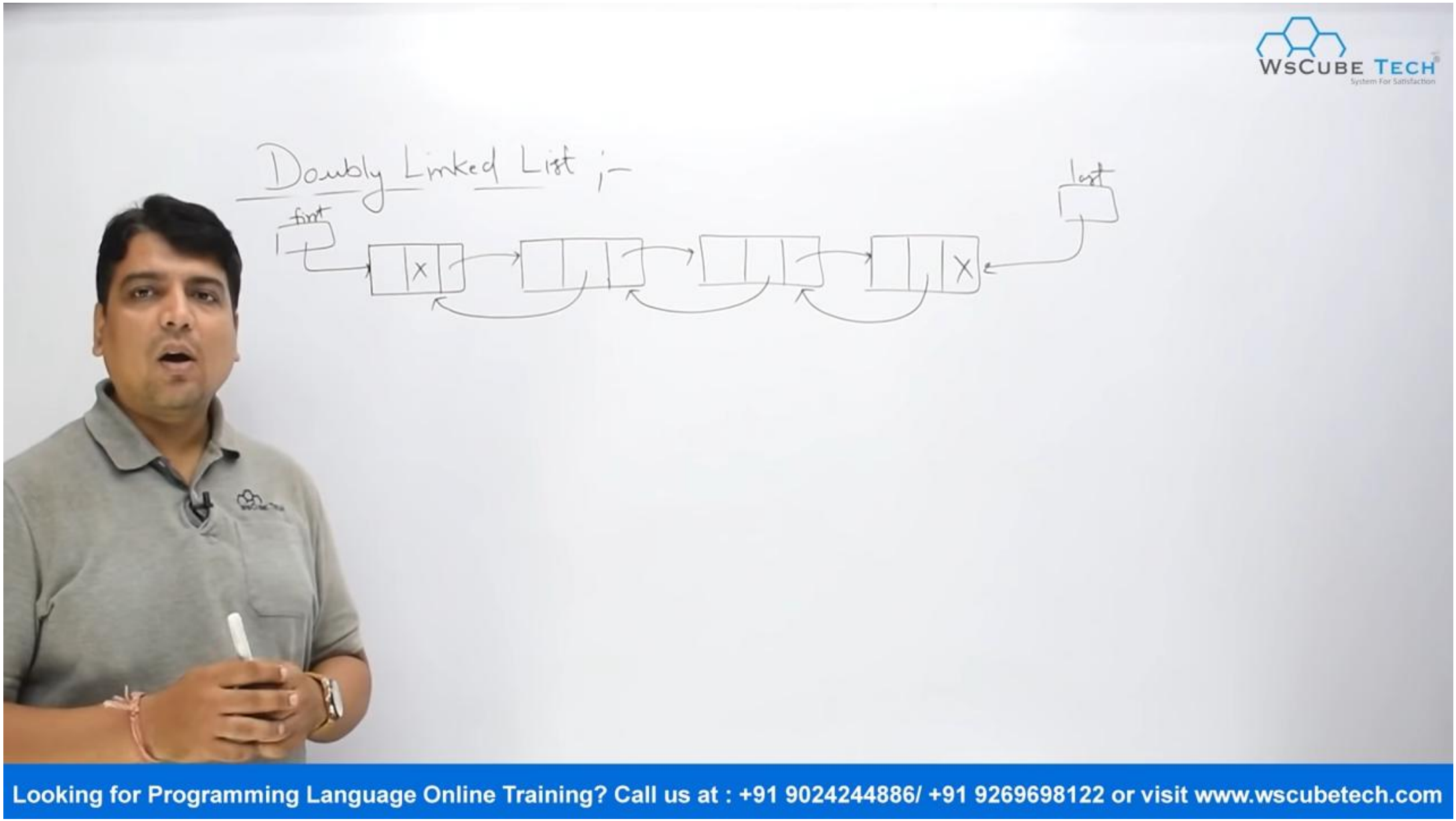
- We can go to any Node
- Save the time to go first to last Node
- All Node Contains Valid Address

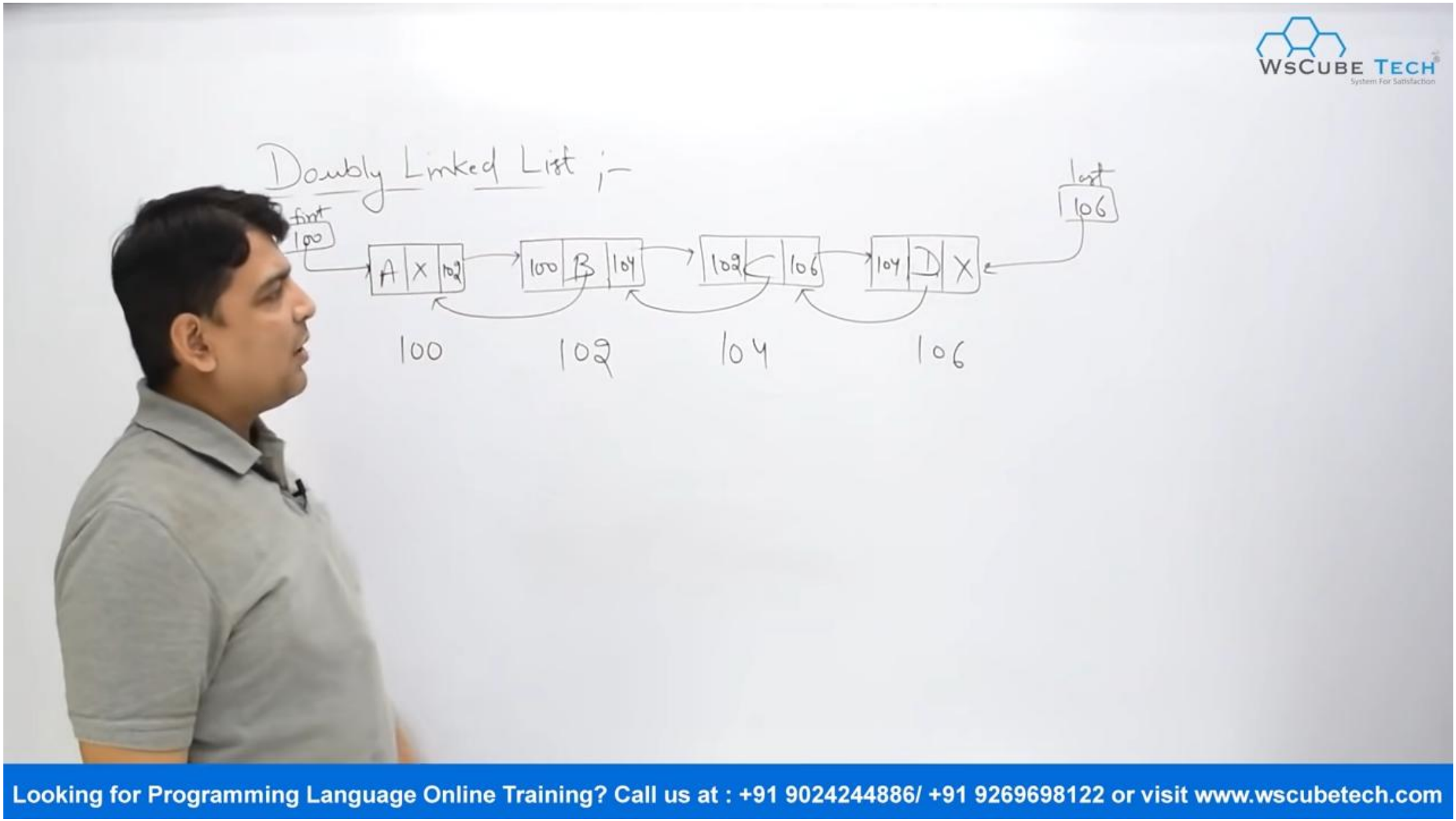
Disadvantages

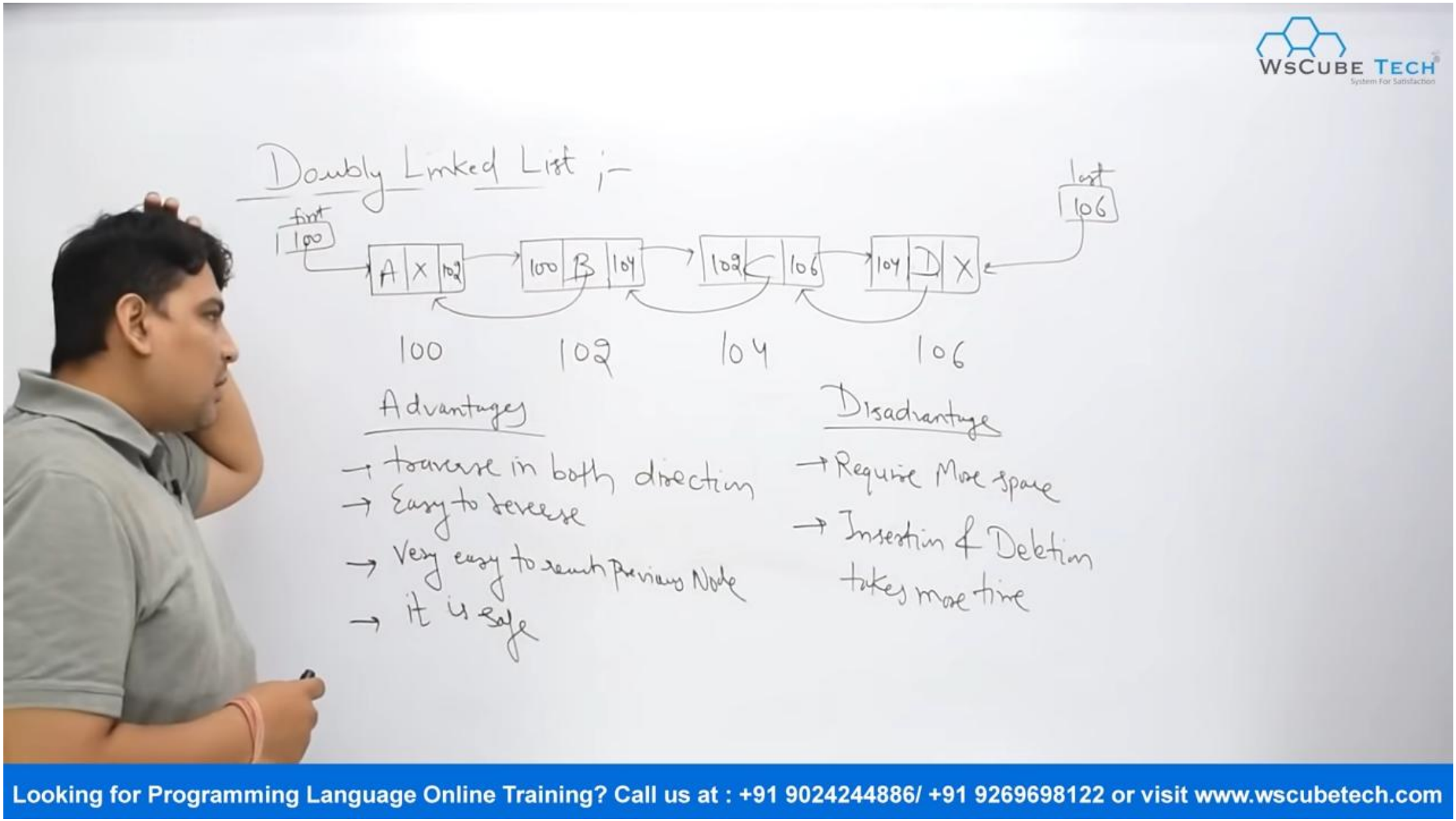
- Not easy to Reverse
- May go to infinite loop

Data Structure & Algorithm

Doubly Linked List







Data Structure & Algorithm

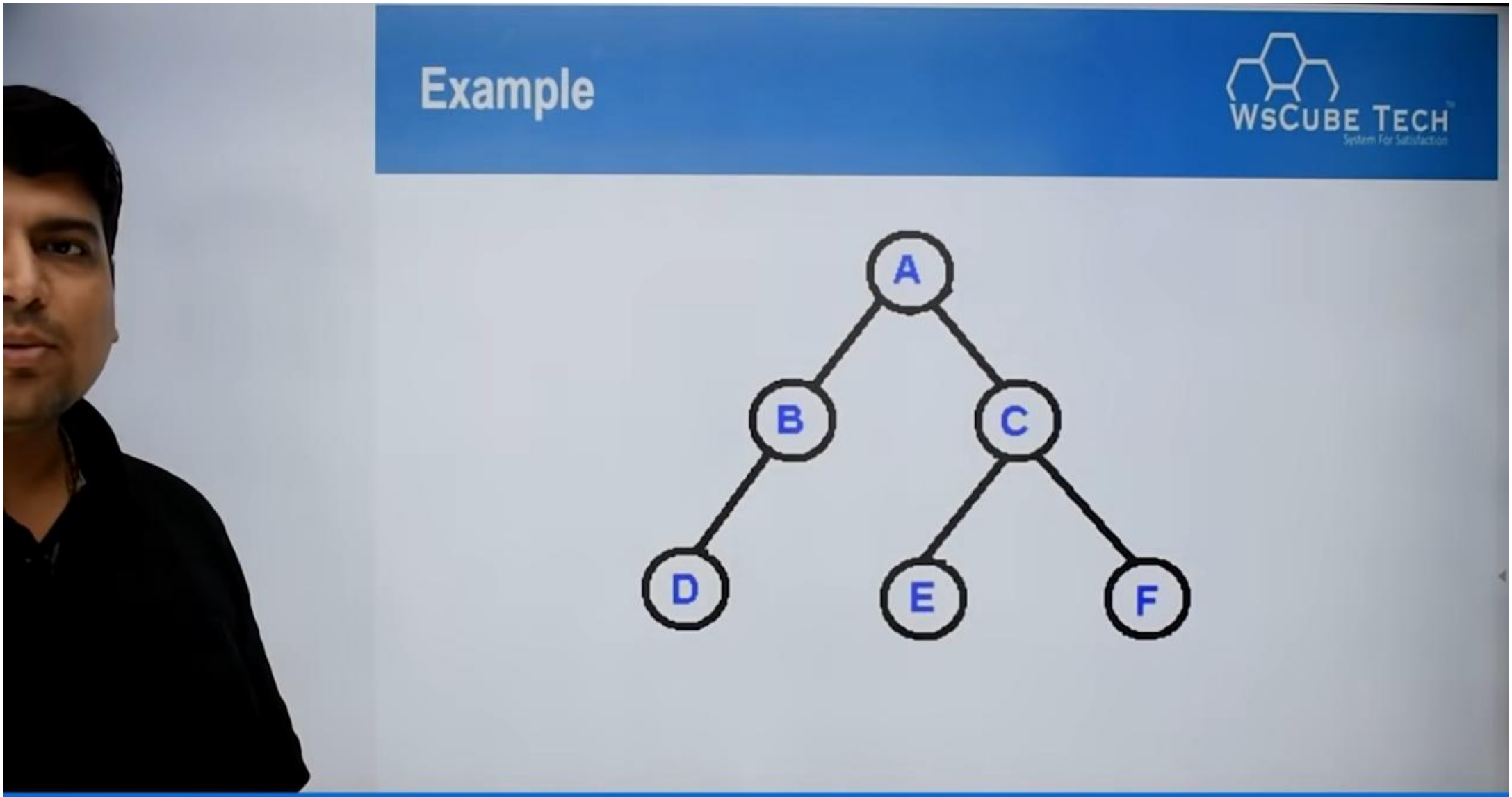
Tree in DSA


www.wscubetech.com
ntroduction to TREE in
Data Structure
1st Floor, Laxmi Tower, Bhaskar Circle, Ratanada, Jodhpur.
Development – Training - Placement - Outsourcing
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

What is Tree Data Structure?

WSCUBE TECH
System For Satisfaction

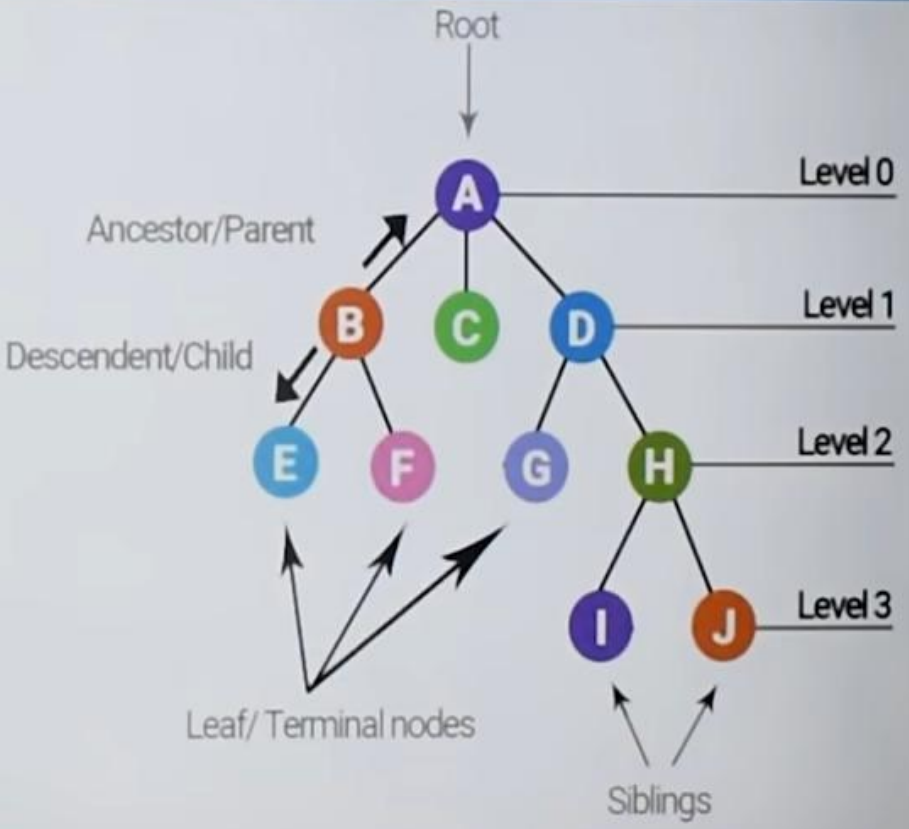
- Tree is a hierarchical data structure and follow parent child relationship.
- It is a non-linear data structure.
- A tree is a collection of elements called nodes. Each node contains some value or element.
- The node at the top of the tree is called **root** or the origin of tree.
- It represents the nodes connected by edges.



Tree Terminology

WSCUBE TECH
System For Satisfaction

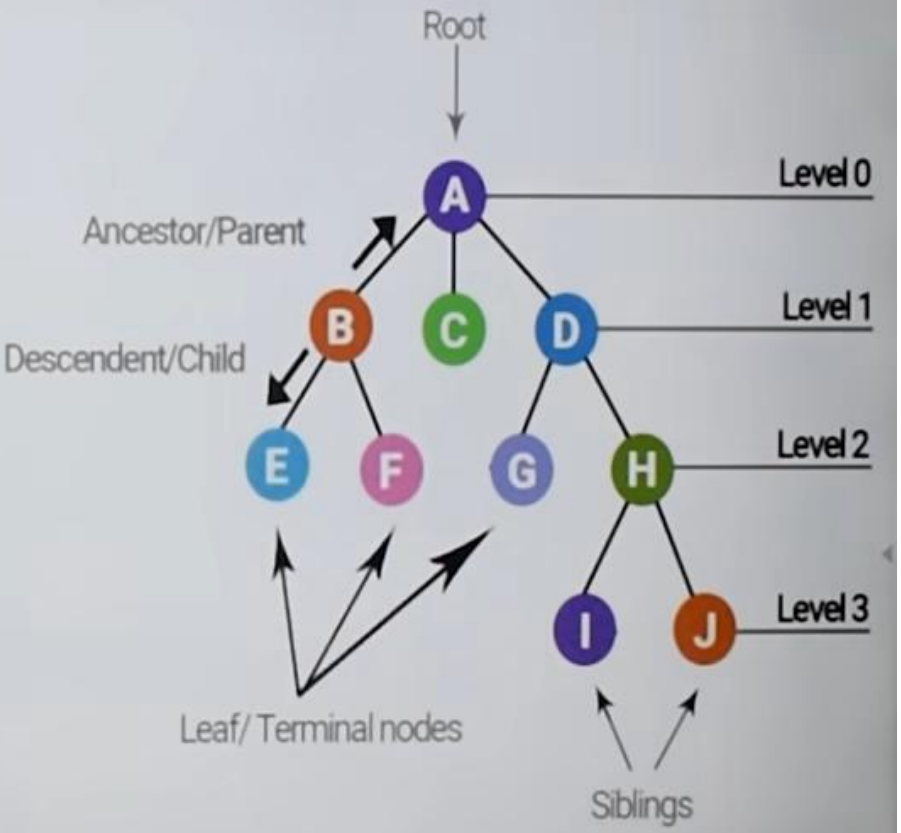
- **Root**: Root is a special node in a tree. The entire tree is referenced through it. It does not have a parent.
- **Parent Node**: Parent node is an immediate predecessor of a node.
- **Child Node**: All immediate successors of a node are its children.
- **Siblings**: Nodes with the same parent are called Siblings.
- **Path**: Path is a number of successive edges from source node to destination node.



Tree Terminology Continued...

WsCube Tech
System For Satisfaction

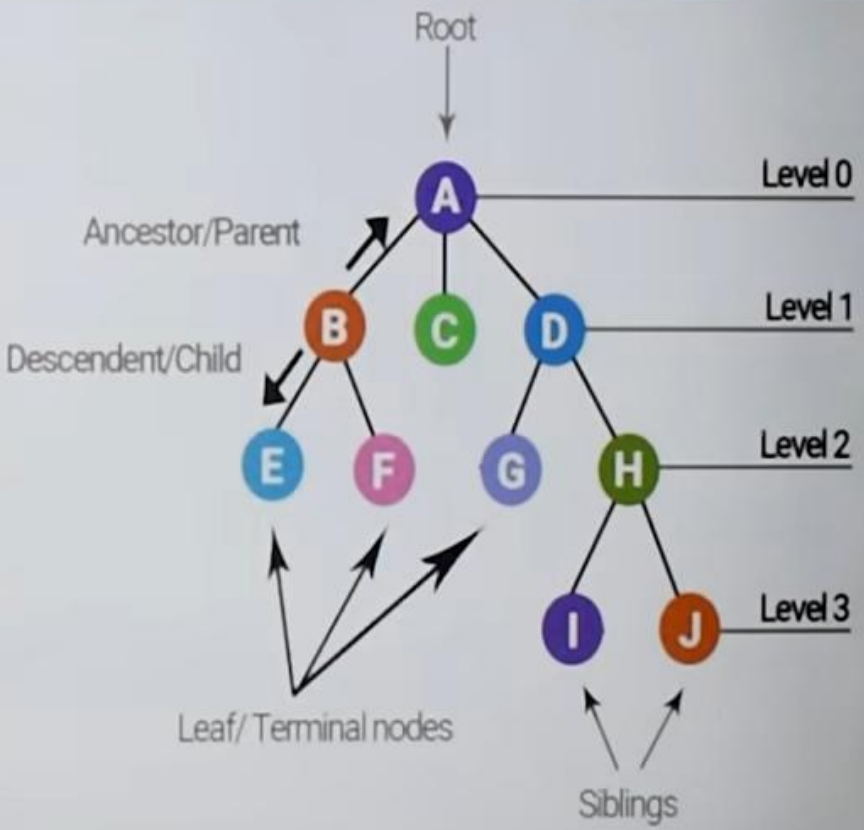
- **Edge:** Edge is a connection between one node to another. It is a line between two nodes or a node and a leaf.
- **Leaf**: The node which does not have any child node is called the leaf node. Leaf nodes are also called as external nodes or terminal nodes.
- **Levels**: Level of a node represents the generation of a node. If the root node is at level 0, then its next child node is at level 1, its grandchild is at level 2, and so on.



Tree Terminology Continued...

WSCUBE TECH
System For Satisfaction

- **Degree of Node**: Degree of a node represents a number of children of a node.
- **Degree of a Tree** is the maximum degree of a node in the tree.
- **Internal Node:**
  - The node which has at least one child is called as an internal node
  - Internal nodes are also called as non-terminal nodes
  - Every non-leaf node is an internal n

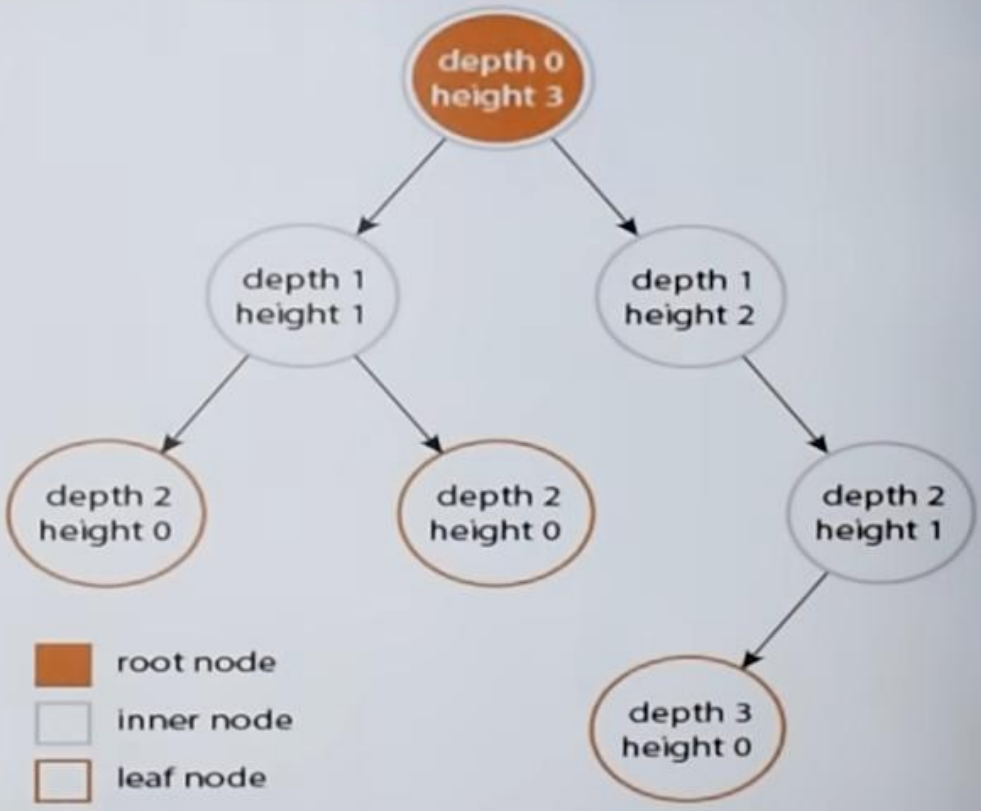


Tree Terminology Continued...

WSCUBE TECH
System For Satisfaction

The depth of a node is the number of edges from the node to the tree's root node.
A root node will have a depth of 0. Note that the depth of the root is 0.

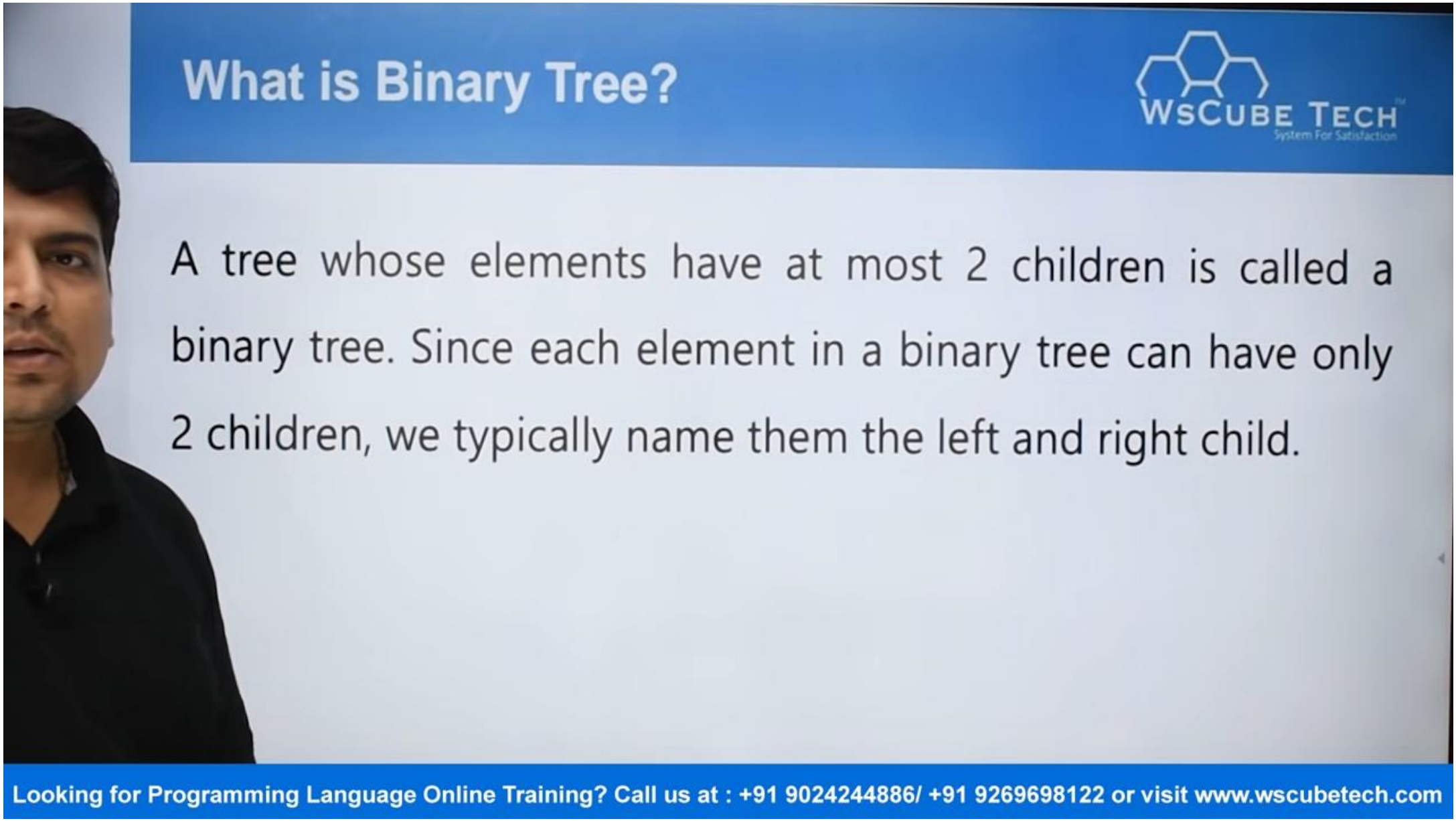
The height of a node is the number of edges on the longest path from the node to a leaf.
A leaf node will have a height of 0. The height of a tree would be the height of its root node.



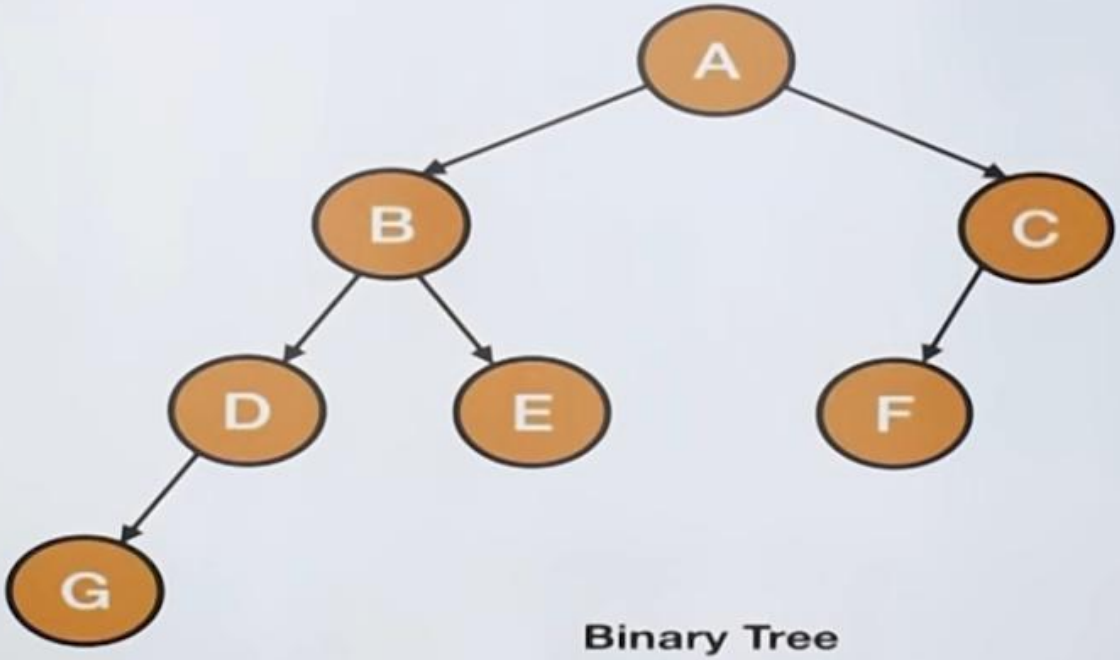
Data Structure & Algorithm

Binary Tree in DSA

WsCube Tech™
System For Satisfaction


What is Binary Tree?
WsCube Tech
System For Satisfaction
A tree whose elements have at most 2 children is called a binary tree. Since each element in a binary tree can have only 2 children, we typically name them the left and right child.
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

Example of Binary Tree

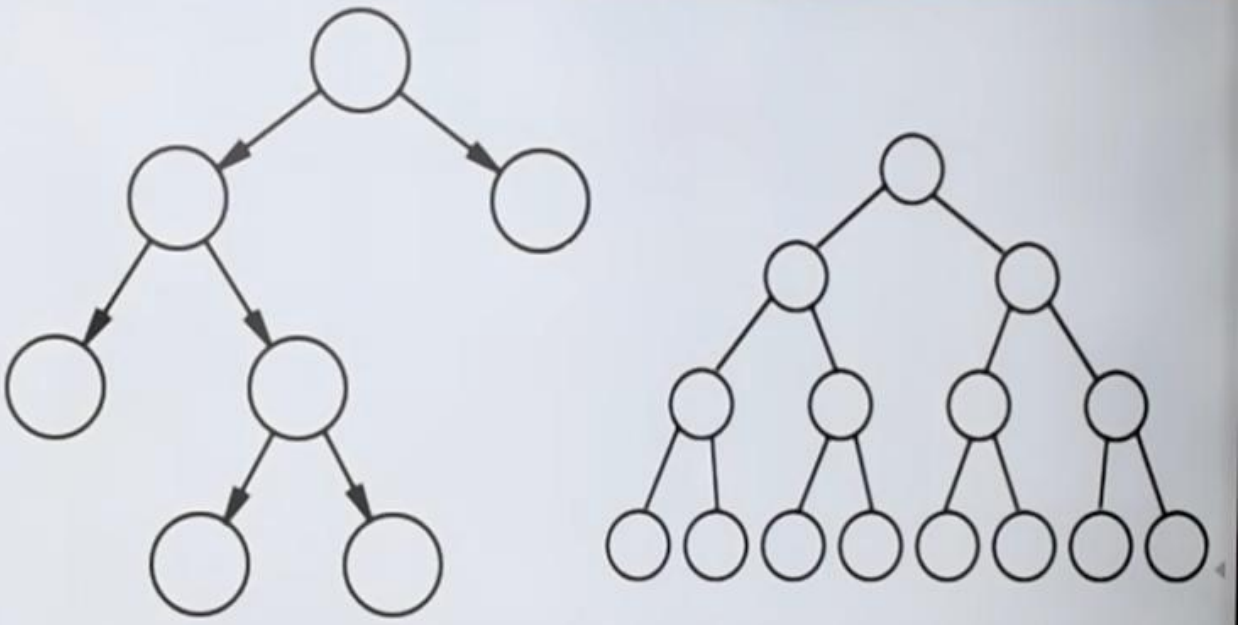


Binary Tree

Full or Strictly Binary Tree

WSCUBE TECH
System For Satisfaction

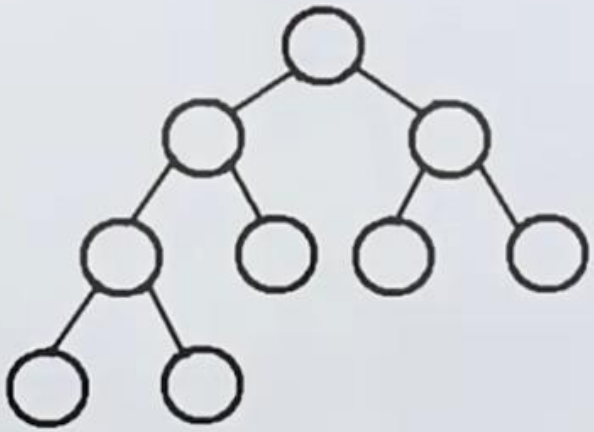
- A binary tree in which every node has either 0 or 2 children is called as a Full binary tree.
- Full binary tree is also called as Strictly binary tree.

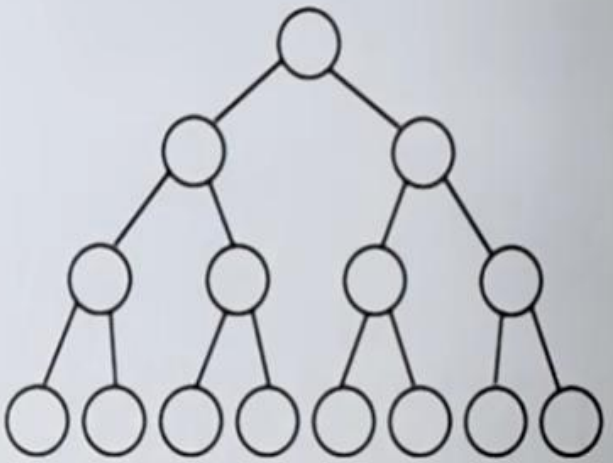


Complete Binary Tree

WSCUBE TECH
System For Satisfaction

A Binary Tree is complete Binary Tree if all levels are completely filled except possibly the last level and the last level has all keys as left as possible

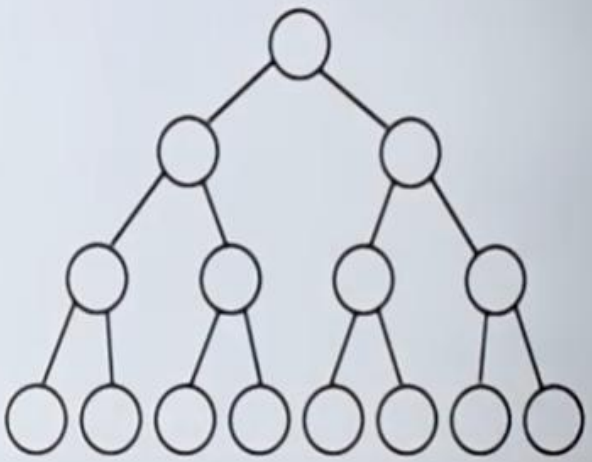




Perfect Binary Tree

WsCube Tech
System For Satisfaction

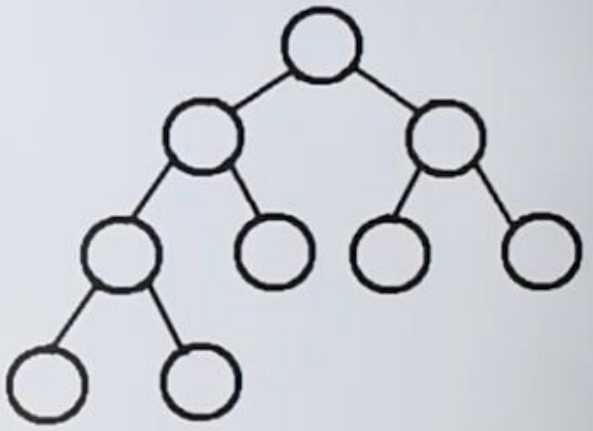
Perfect Binary Tree A Binary tree is Perfect Binary Tree in which all internal nodes have two children and all leaves are at the same level.



Almost Complete Binary Tree

WSCUBE TECH
System For Satisfaction

All leafs at lowest and next to lowest levels only. All except the lowest level is full. No gaps except at the end of a level. A perfectly balanced tree with leaves at the last level all in the leftmost position.



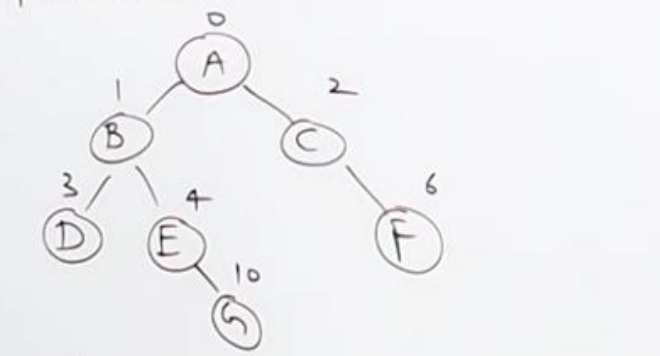
Data Structure & Algorithm

Representation of Binary Tree

WsCube Tech
System For Satisfaction

Representation of Binary Tree :-

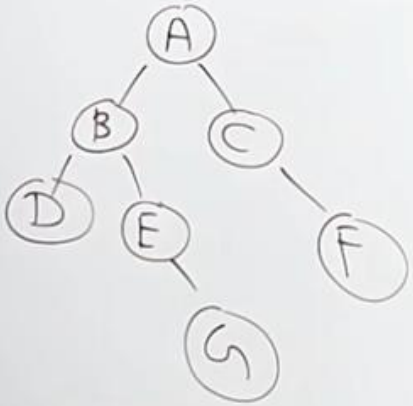
- Sequential Representation or Array Representation
- Linked List Representation

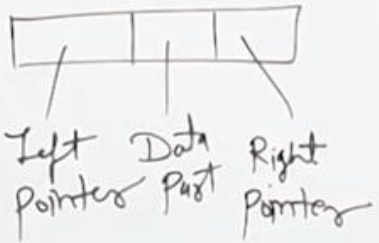


Representation of Binary Tree :-

→ Sequential Representation or Array Representation

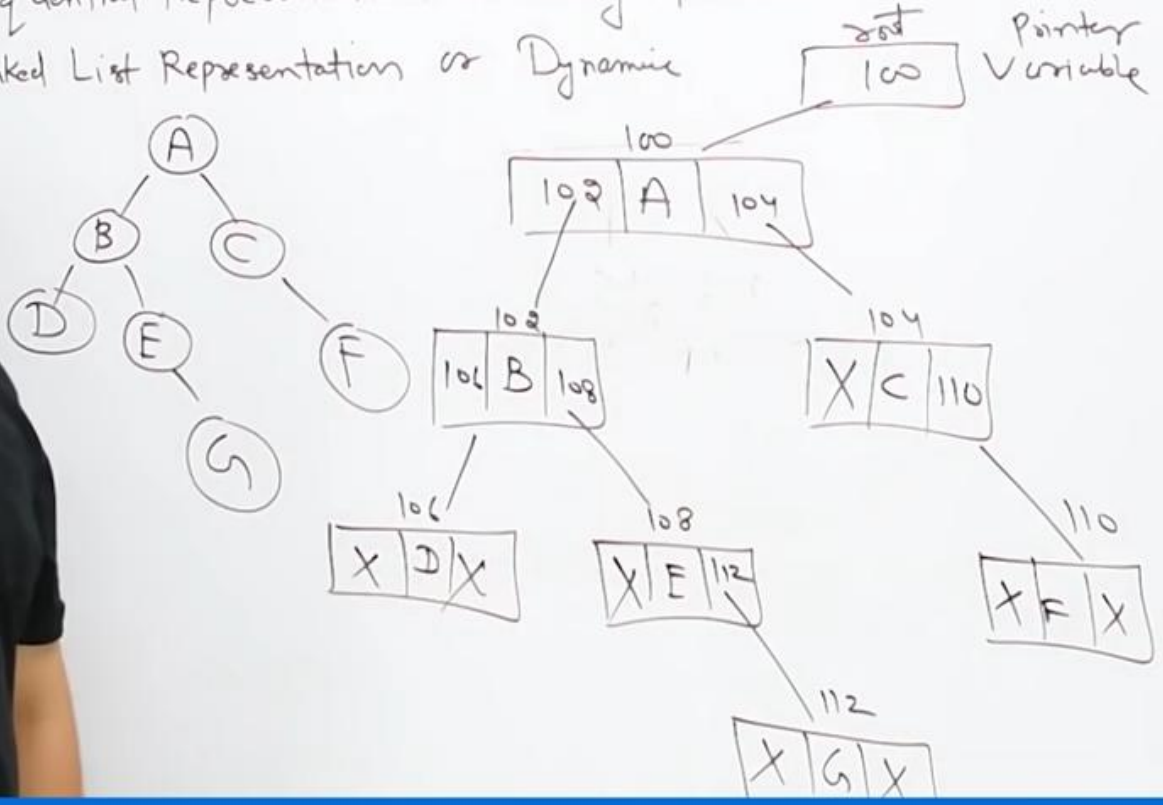
→ Linked List Representation or Dynamic ..





Representation of Binary Tree :-

- Sequential Representation or Array Representation
- Linked List Representation or Dynamic

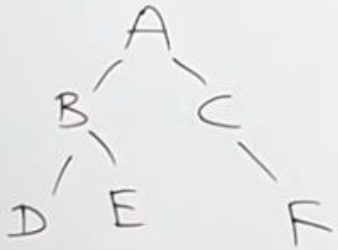


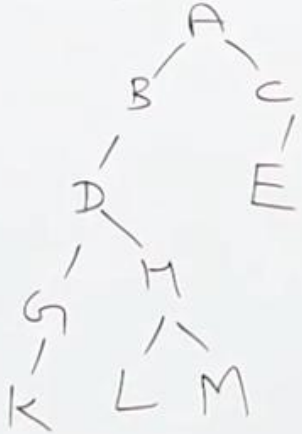
Data Structure & Algorithm

PreOrder Traversals of Binary Tree

Traversing of Binary Tree :-

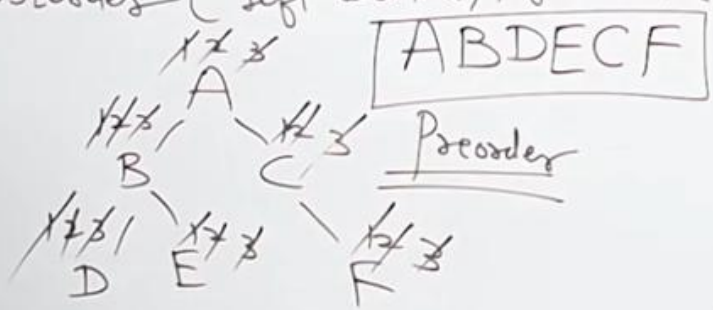
→ Preorder (Root, Left Sub tree, Right Sub tree)
→ Inorder (left Sub tree, Root, Right Sub tree)
→ Postorder (left Sub tree, Right Sub tree, Root)





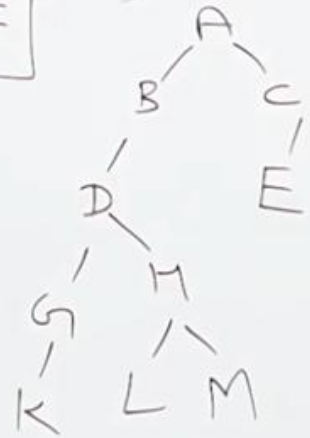
Traversing of Binary Tree :-

→ Preorder (Root, Left Sub tree, Right Sub tree)
→ Inorder (left Sub tree, Root, Right Sub tree)
→ Postorder (left Sub tree, Right Sub tree, Root)

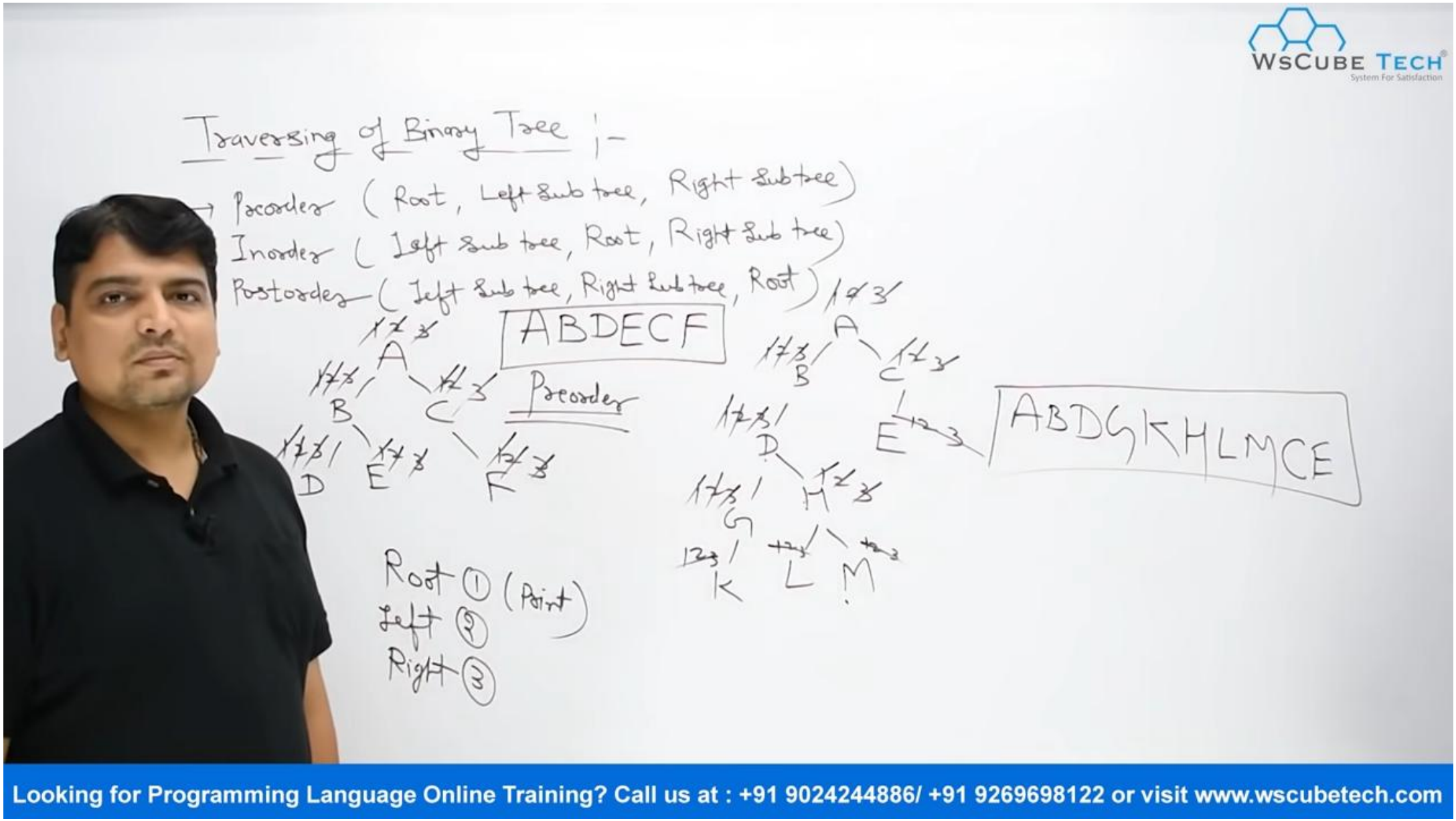


ABDECF

Preorder

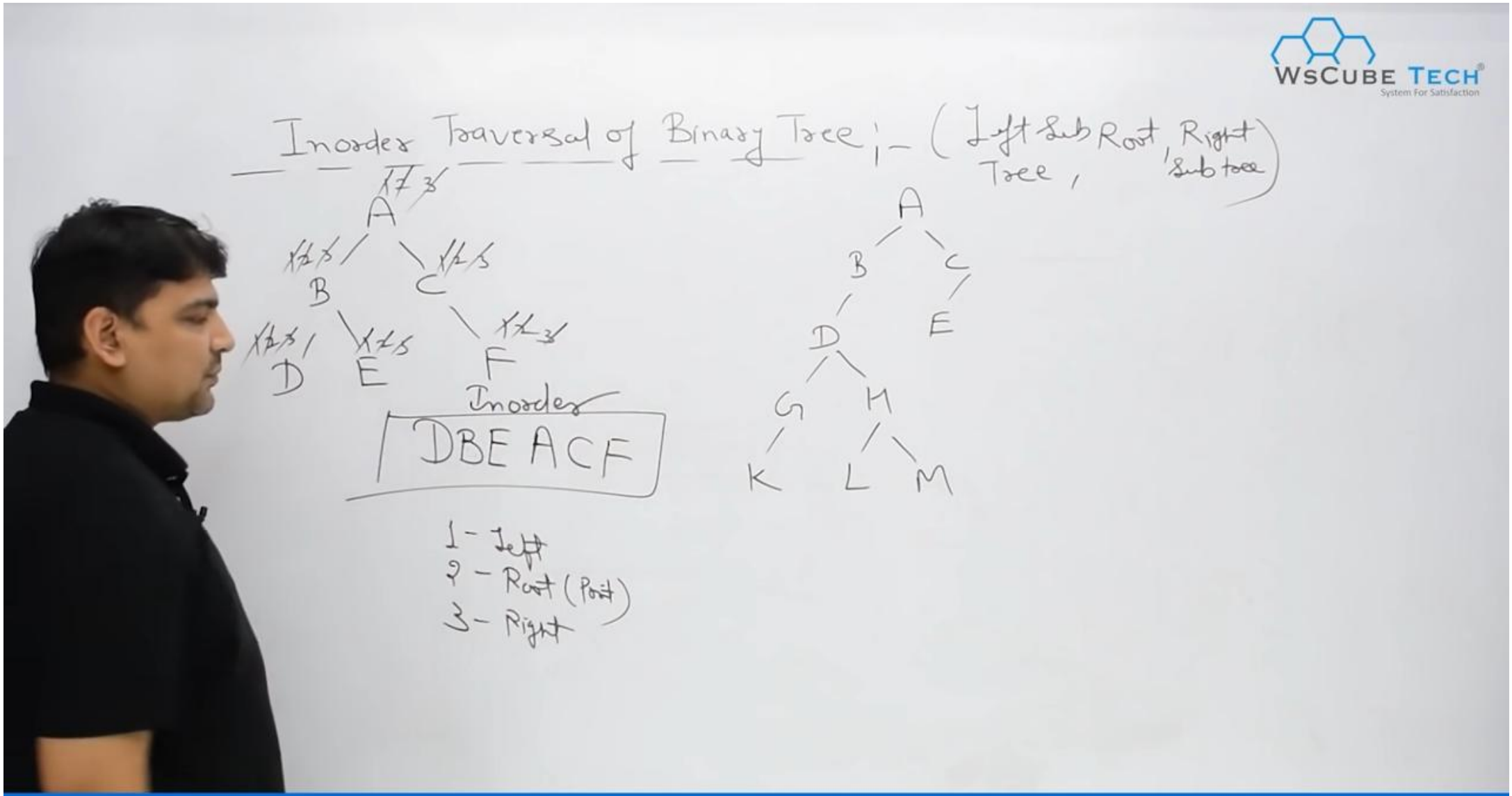


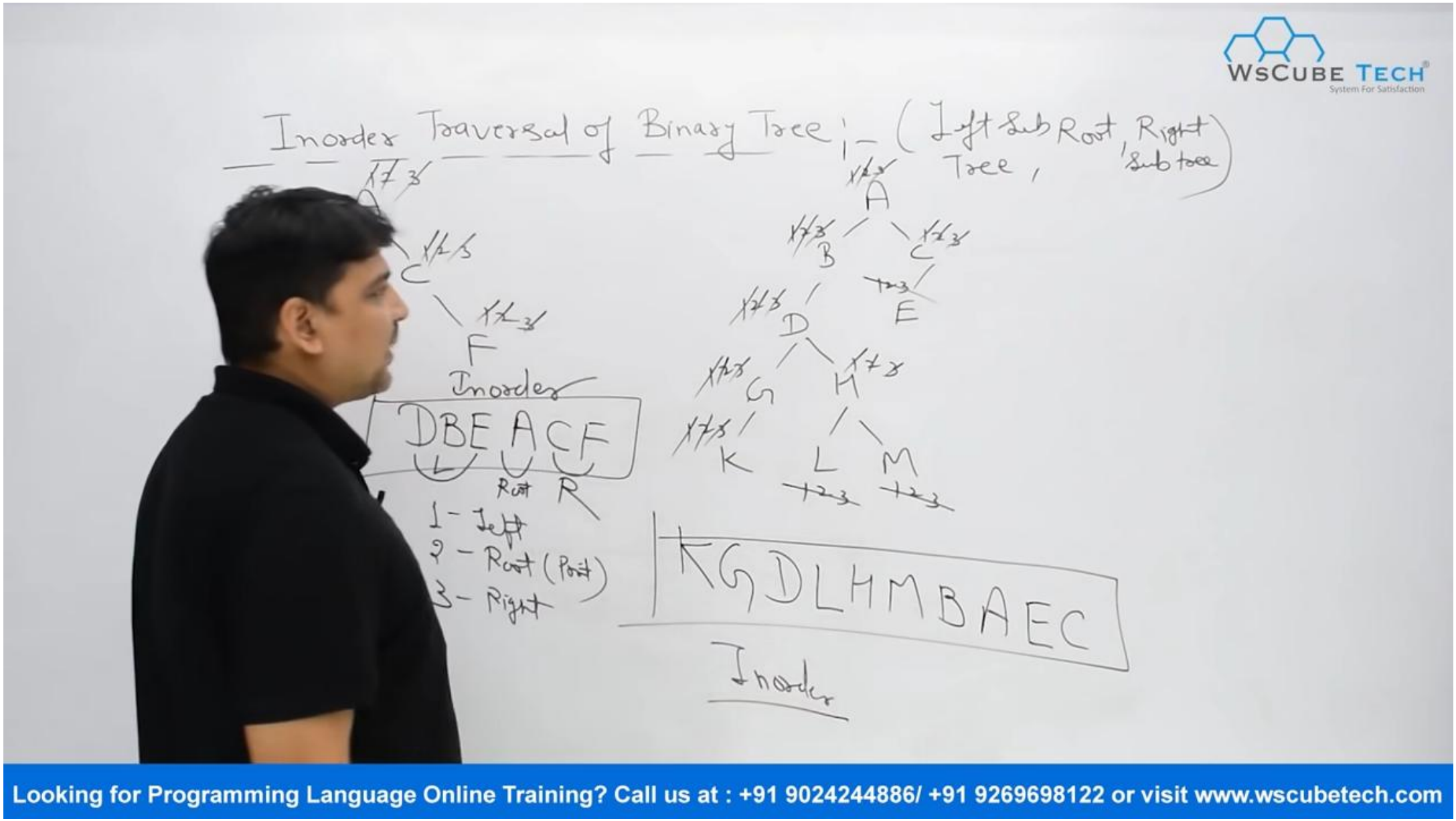
Root ① (Print)
Left ②
Right ③



Data Structure & Algorithm

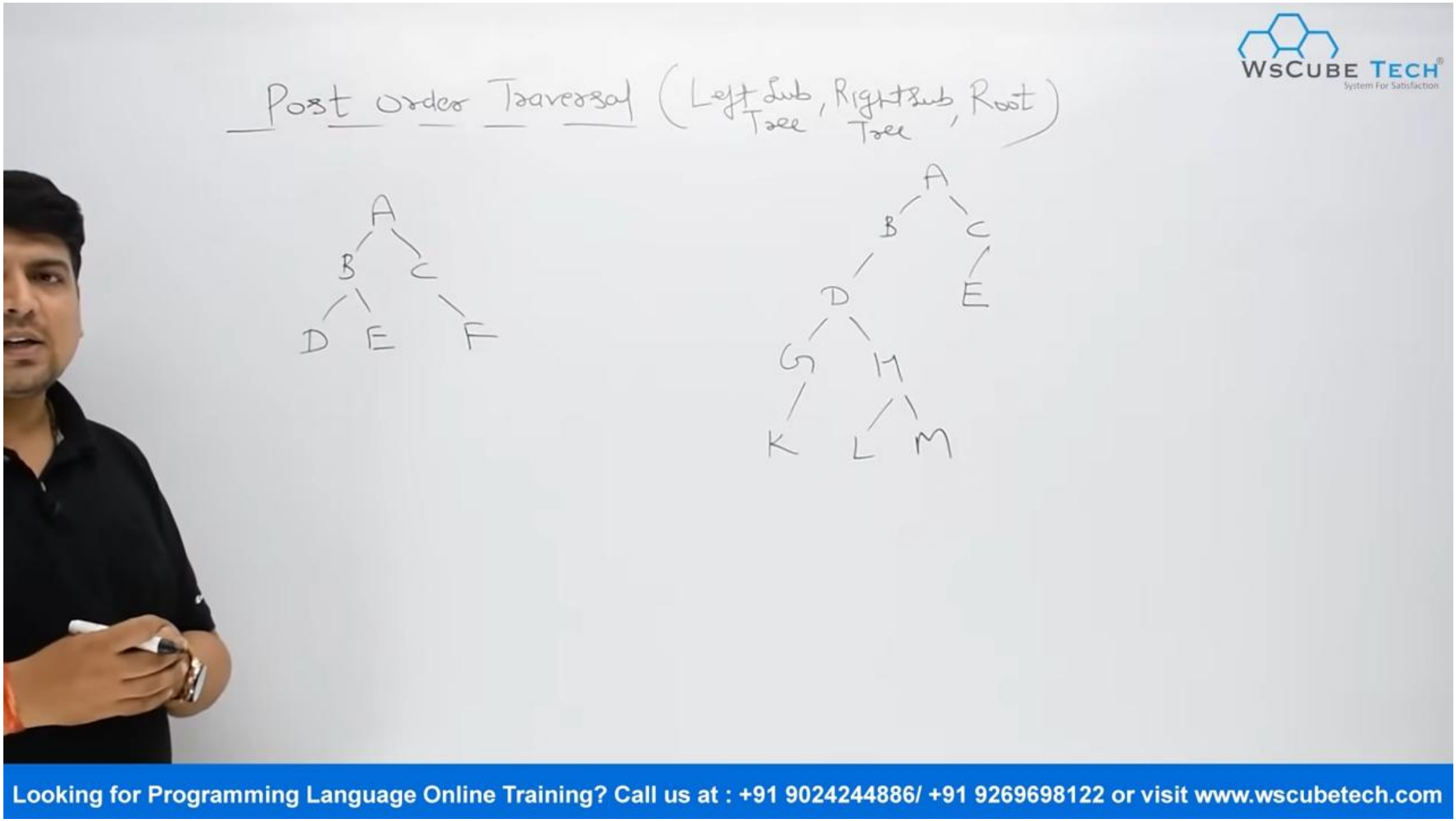
Inorder Traversal

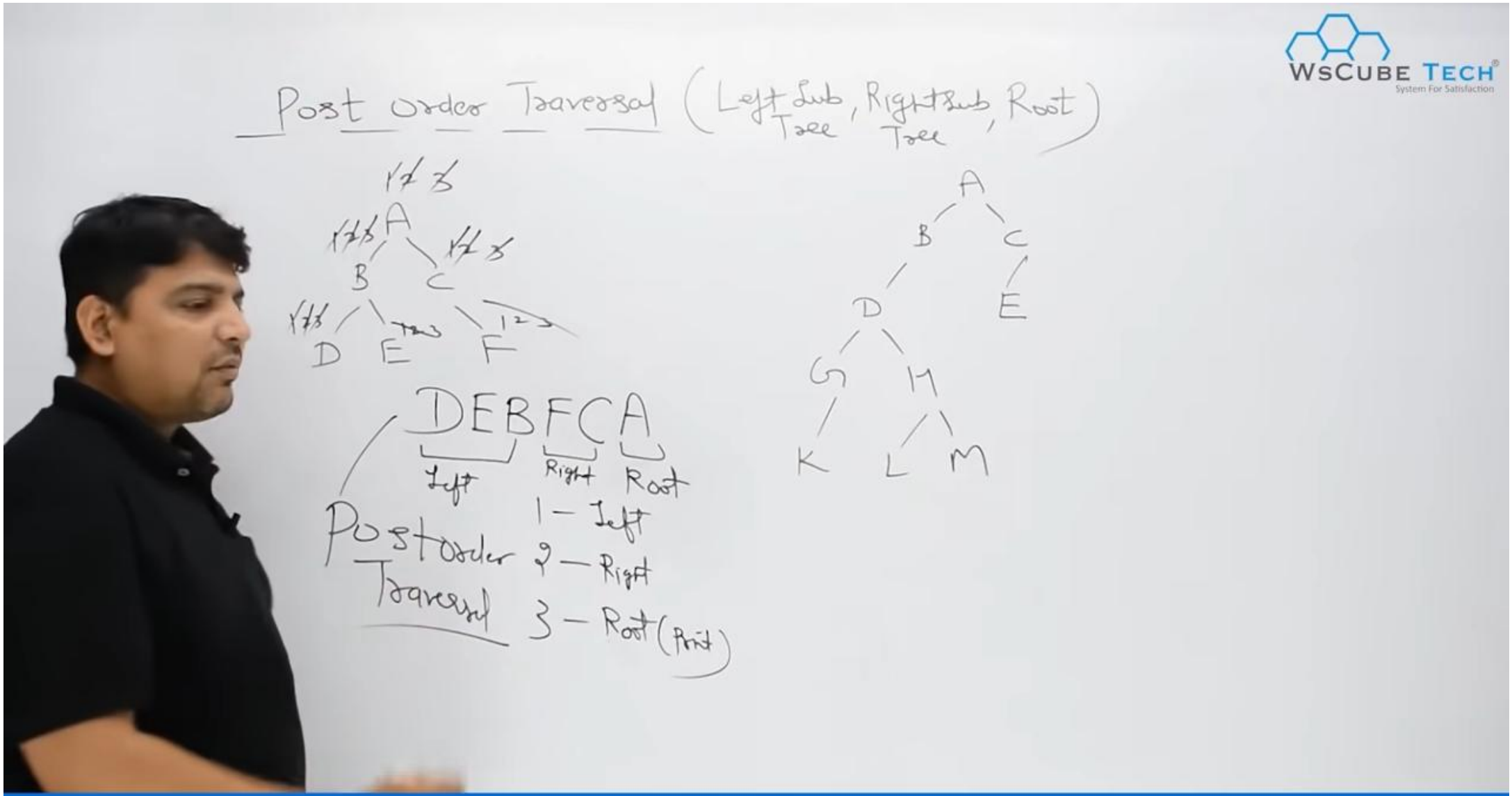


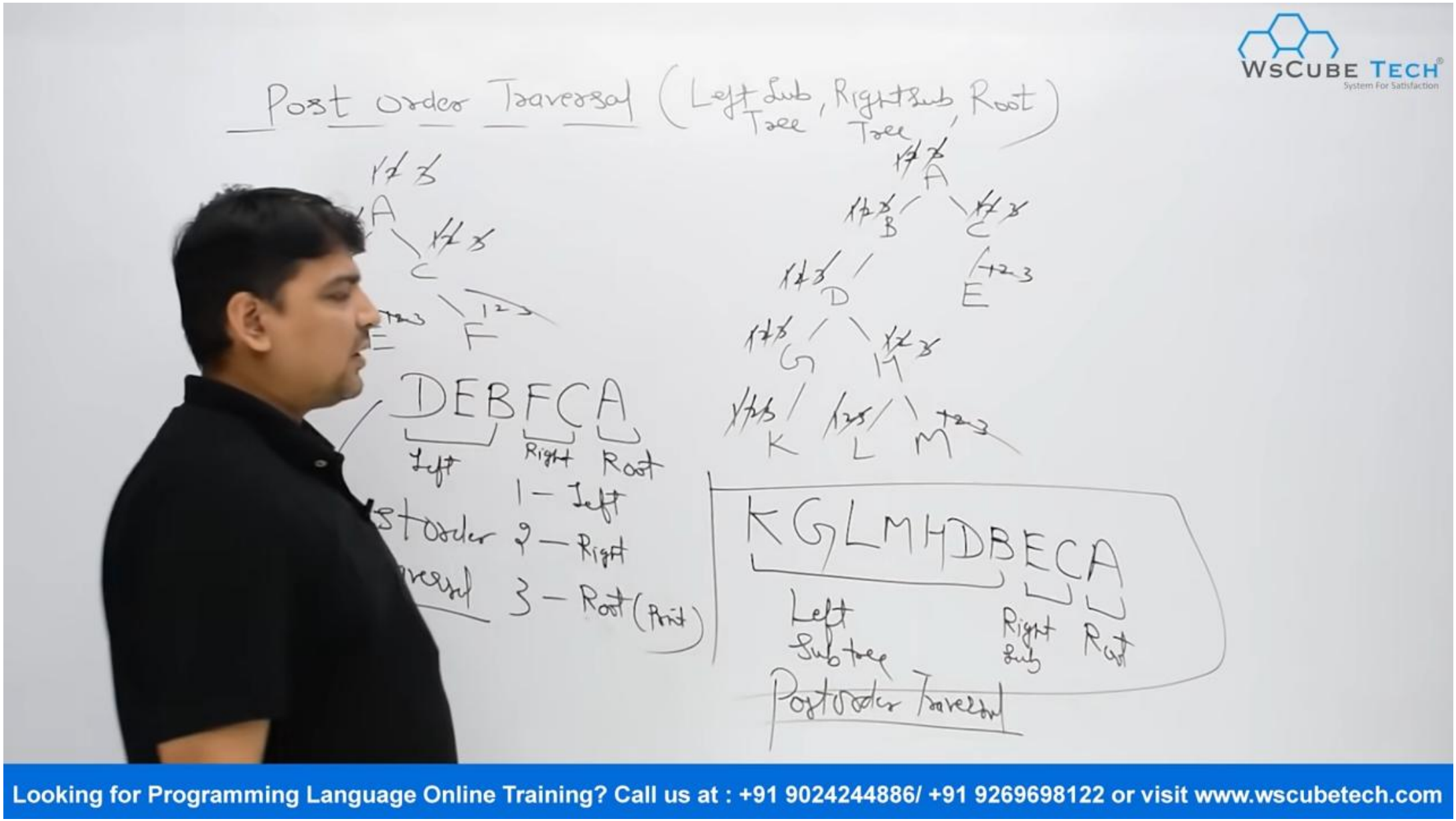


Data Structure & Algorithm

Post Order Traversal of Binary Tree





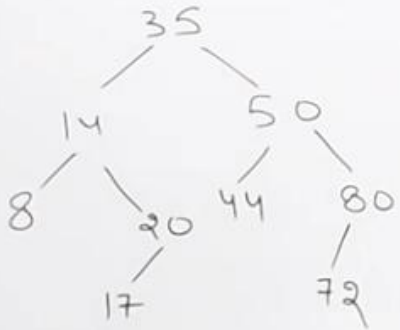


Data Structure & Algorithm

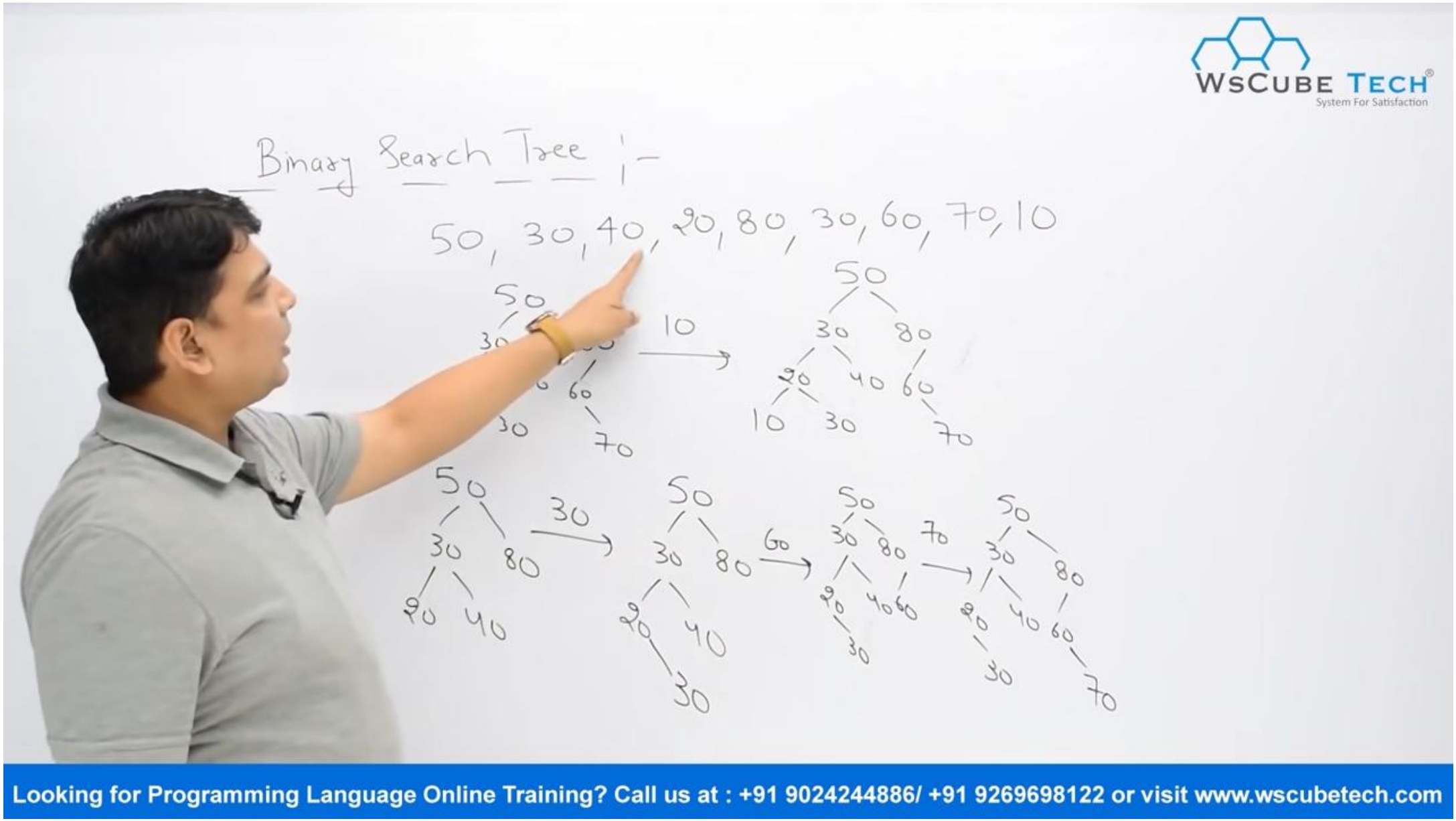
Binary Search Tree

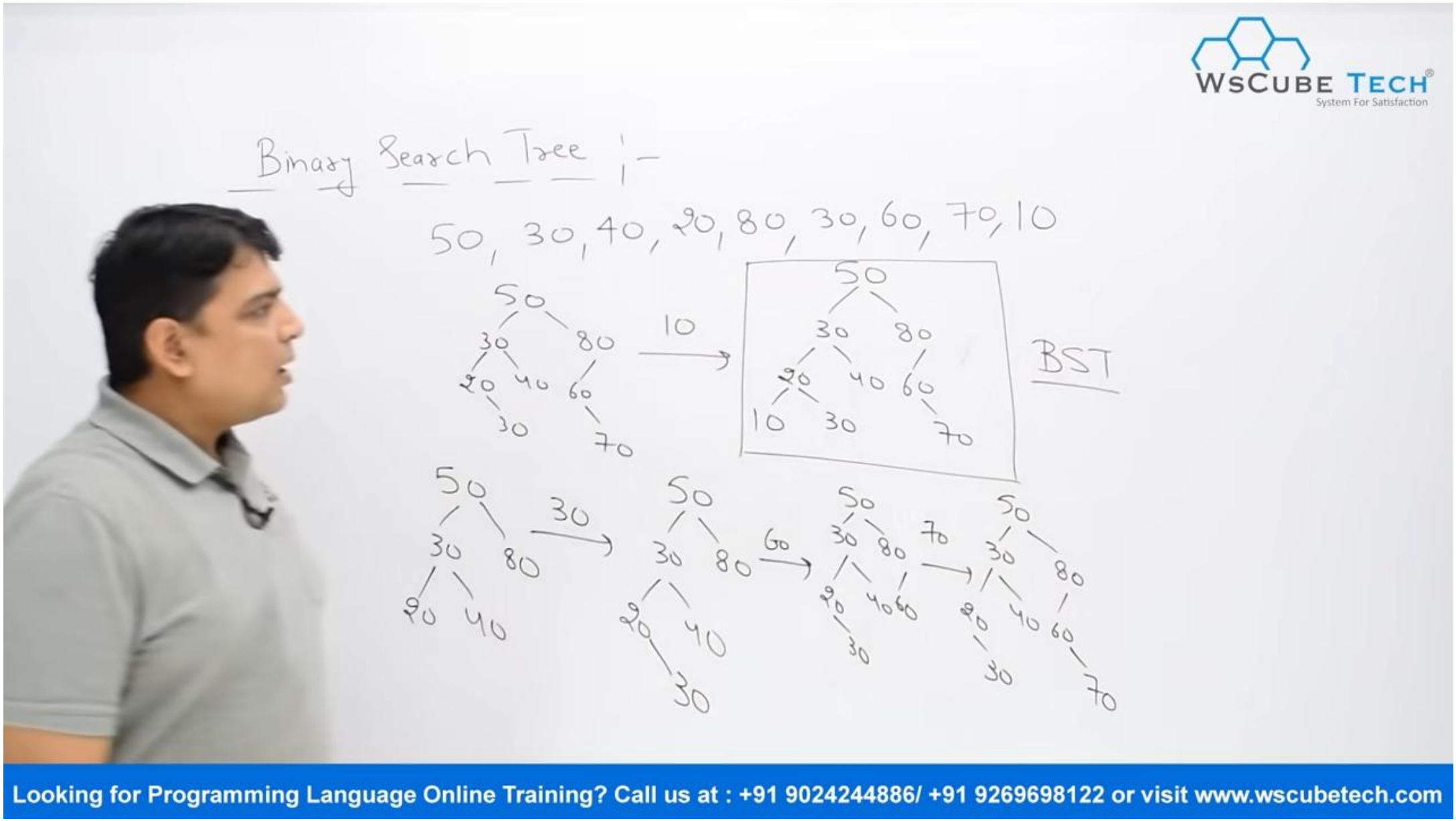
WsCube Tech
System For Satisfaction

Binary Search Tree :-



Left ≤ Data < Right

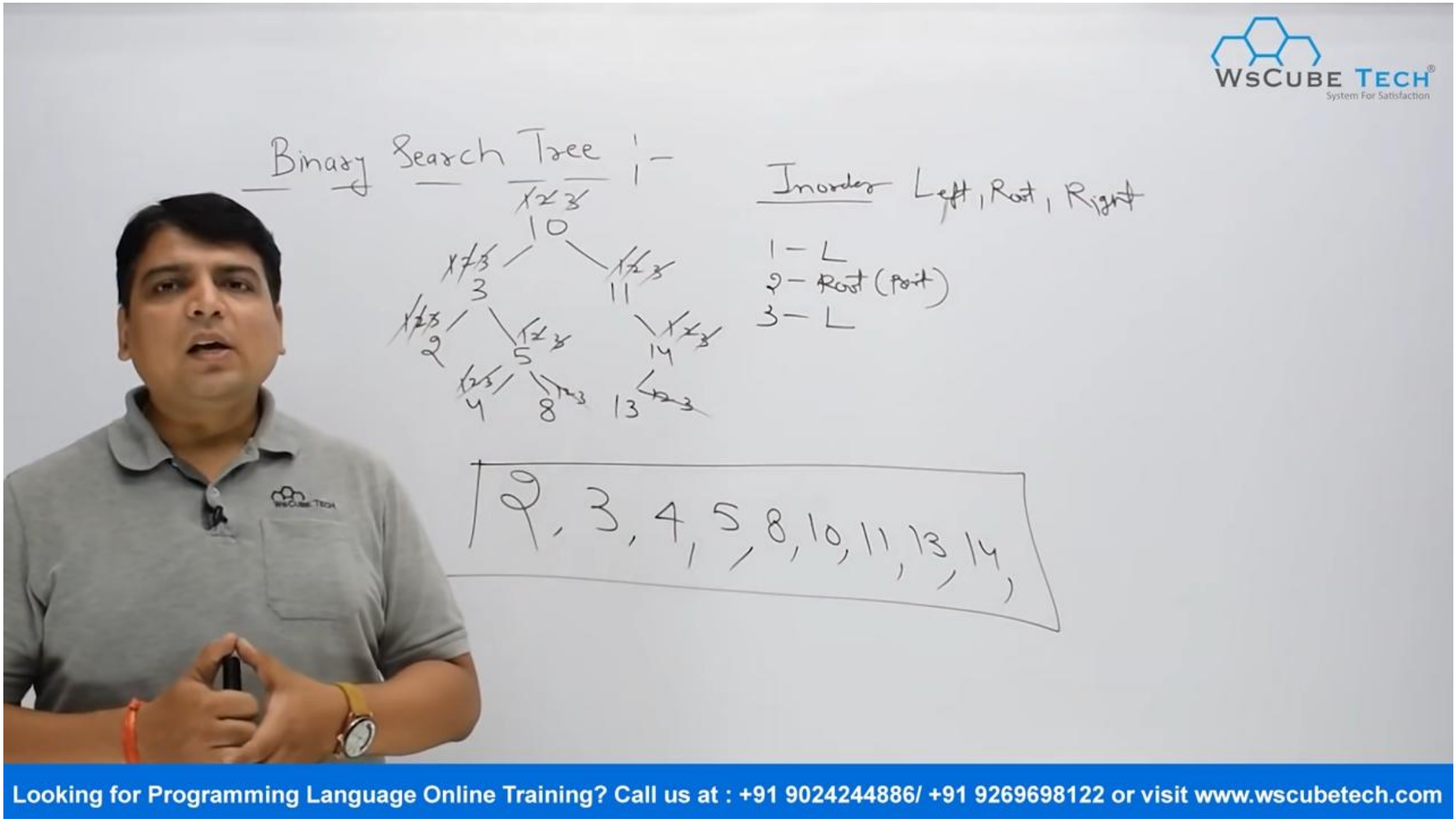


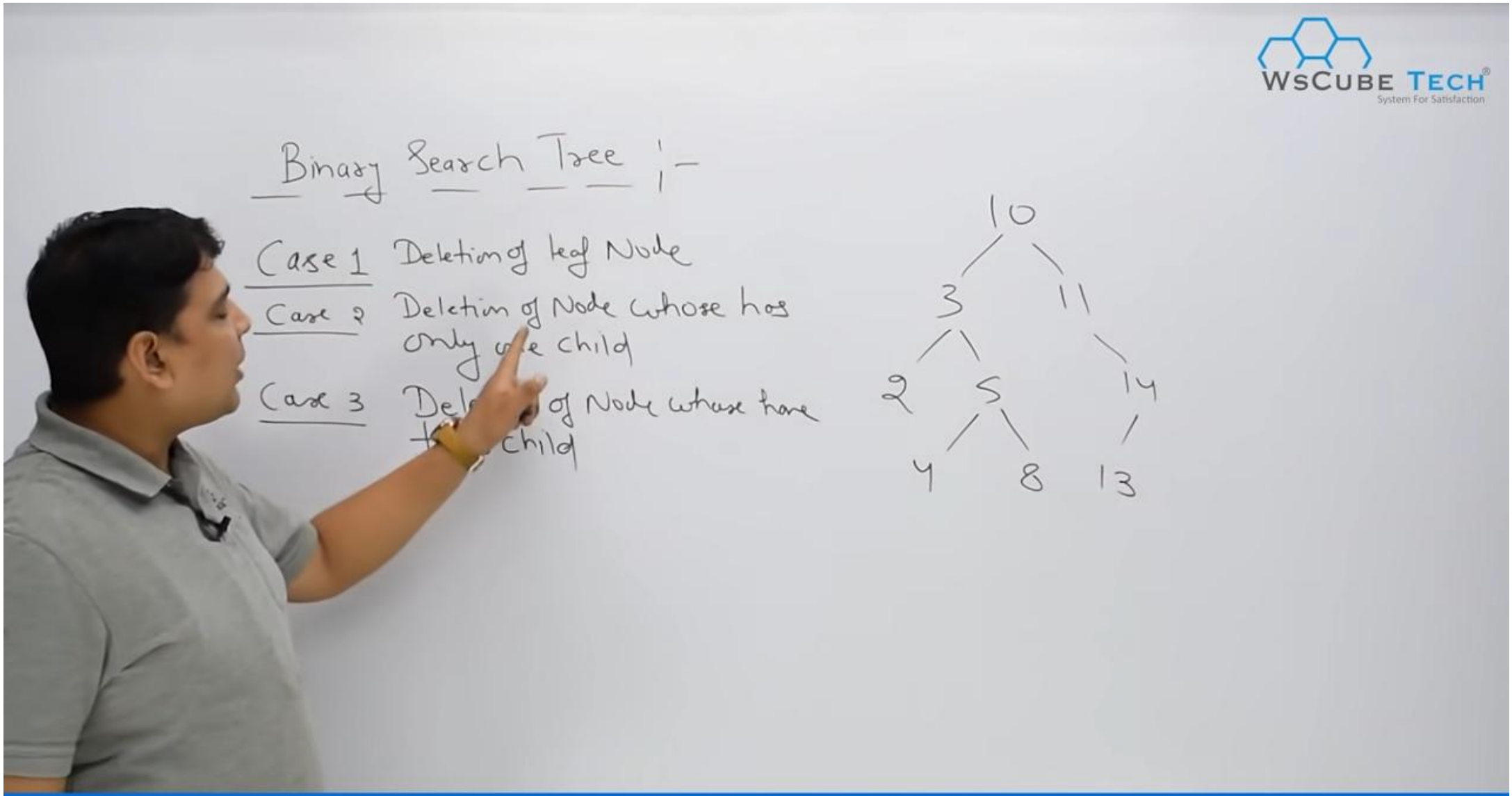

WsCube Tech
System For Satisfaction
Binary Search Tree :-
50, 30, 40, 20, 80, 30, 60, 70, 10
10
BST
30
60
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

Data Structure & Algorithm

Deletion into Binary Search Tree

WsCube Tech™
System For Satisfaction





Data Structure & Algorithm

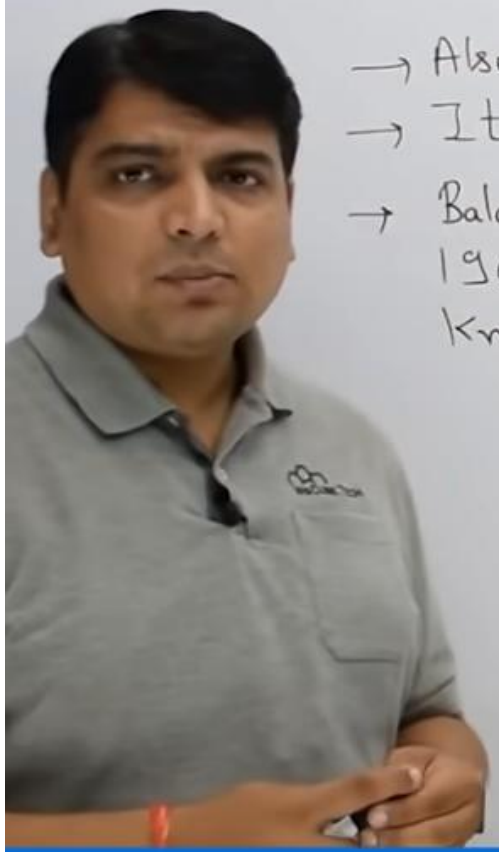
AVL Tree in DSA

WsCube Tech™
System For Satisfaction

Introduction to AVL Tree :-

- → Also called as height balance Tree
- → It is binary Search Tree
- → Balance Tree's Concept was introduced in 1962 by Adelson-Velski-Landis known as AVL Tree.

$$\text{Balance factor} = h(T^L) - h(T^R)$$



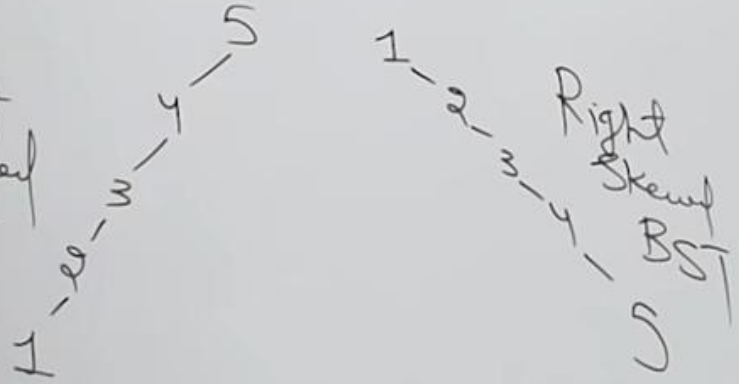
Introduction to AVL Tree :-

→ Also called as height balance Tree

→ It is binary Search Tree

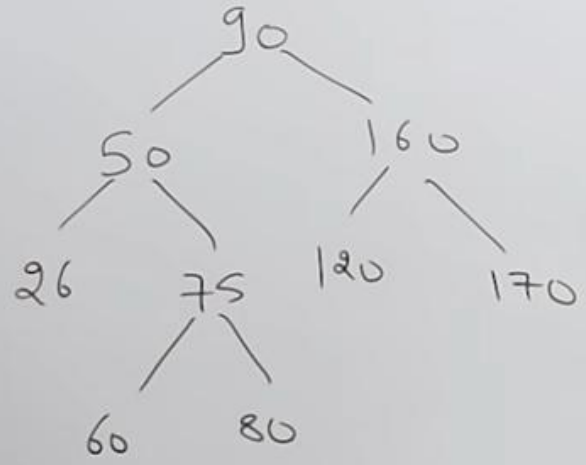
→ Balance Tree's Concept was introduced in 1962 by Adelson-Velskil-Landis known as AVL Tree.

Balance factor = $h(T^L) - h(T^R)$



WSCUBE TECH

Introduction to AVL Tree :-

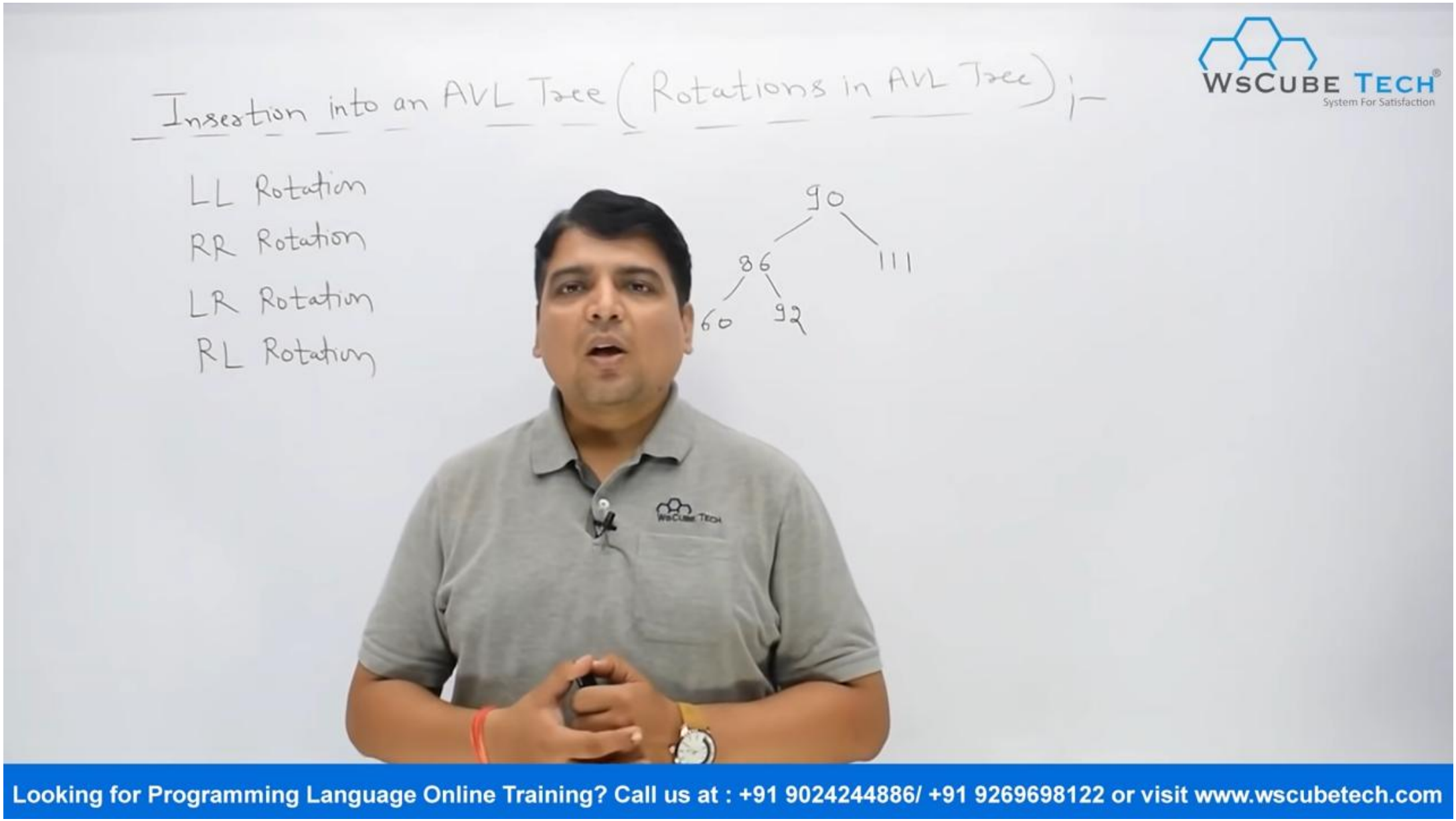


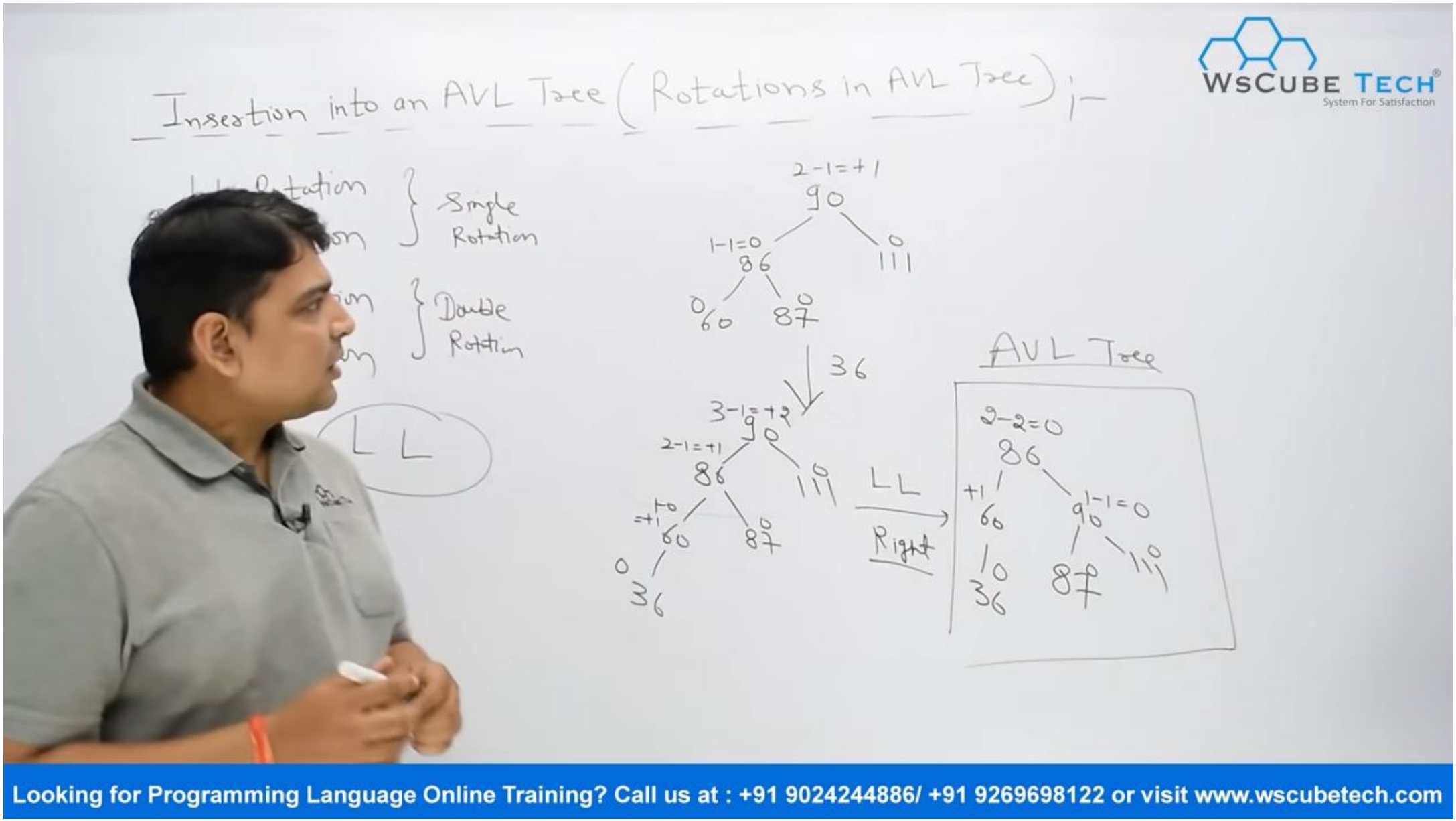
Balance Factor = $h(T^L) - h(T^R)$

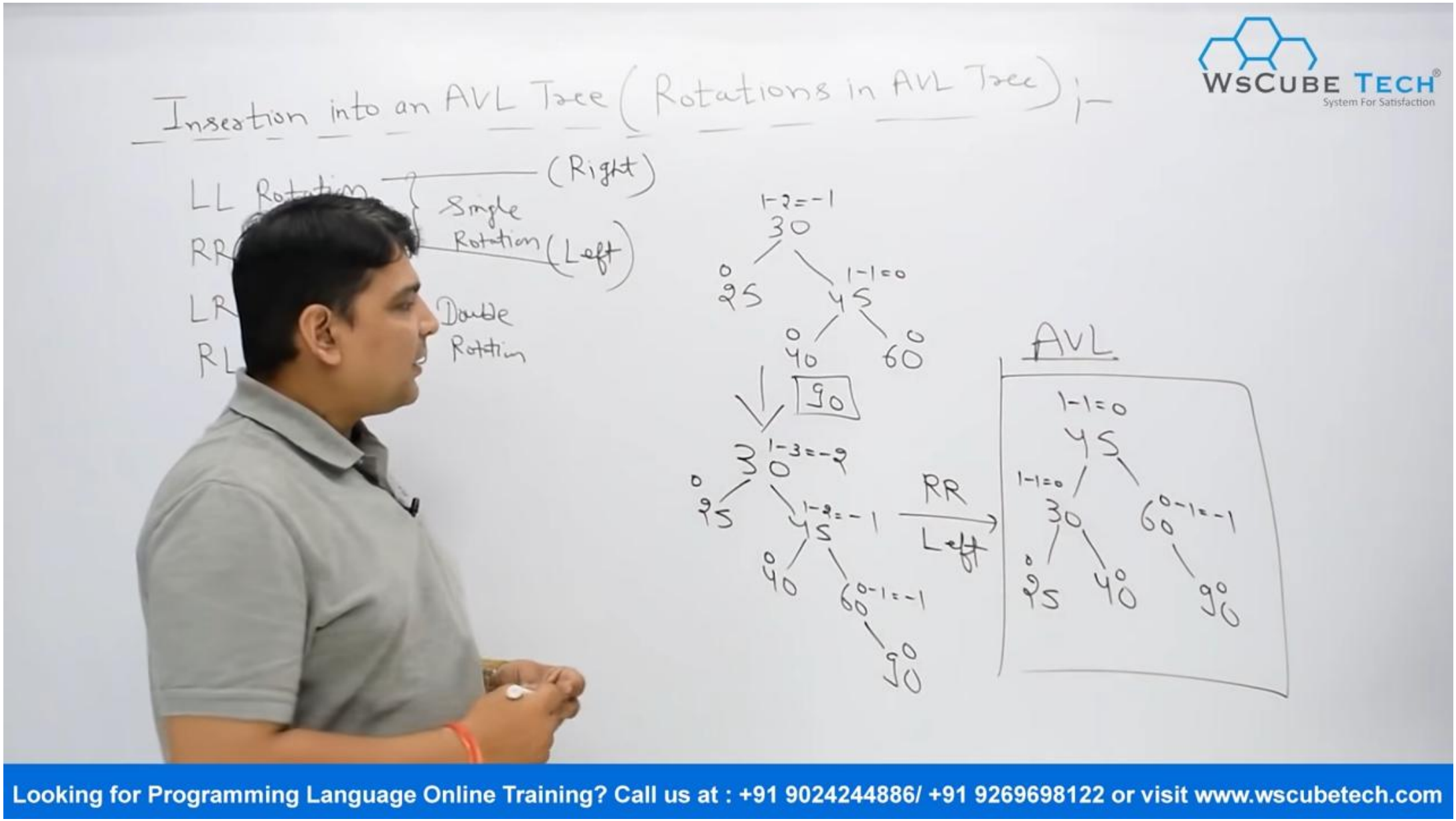
0, +1, -1

Data Structure & Algorithm

AVL Tree Insertion





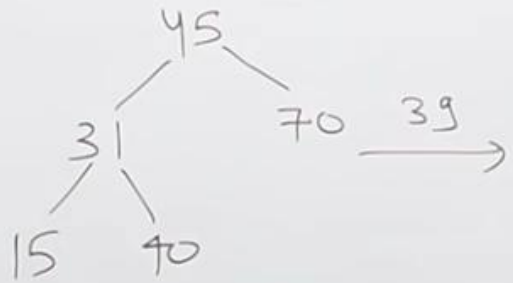


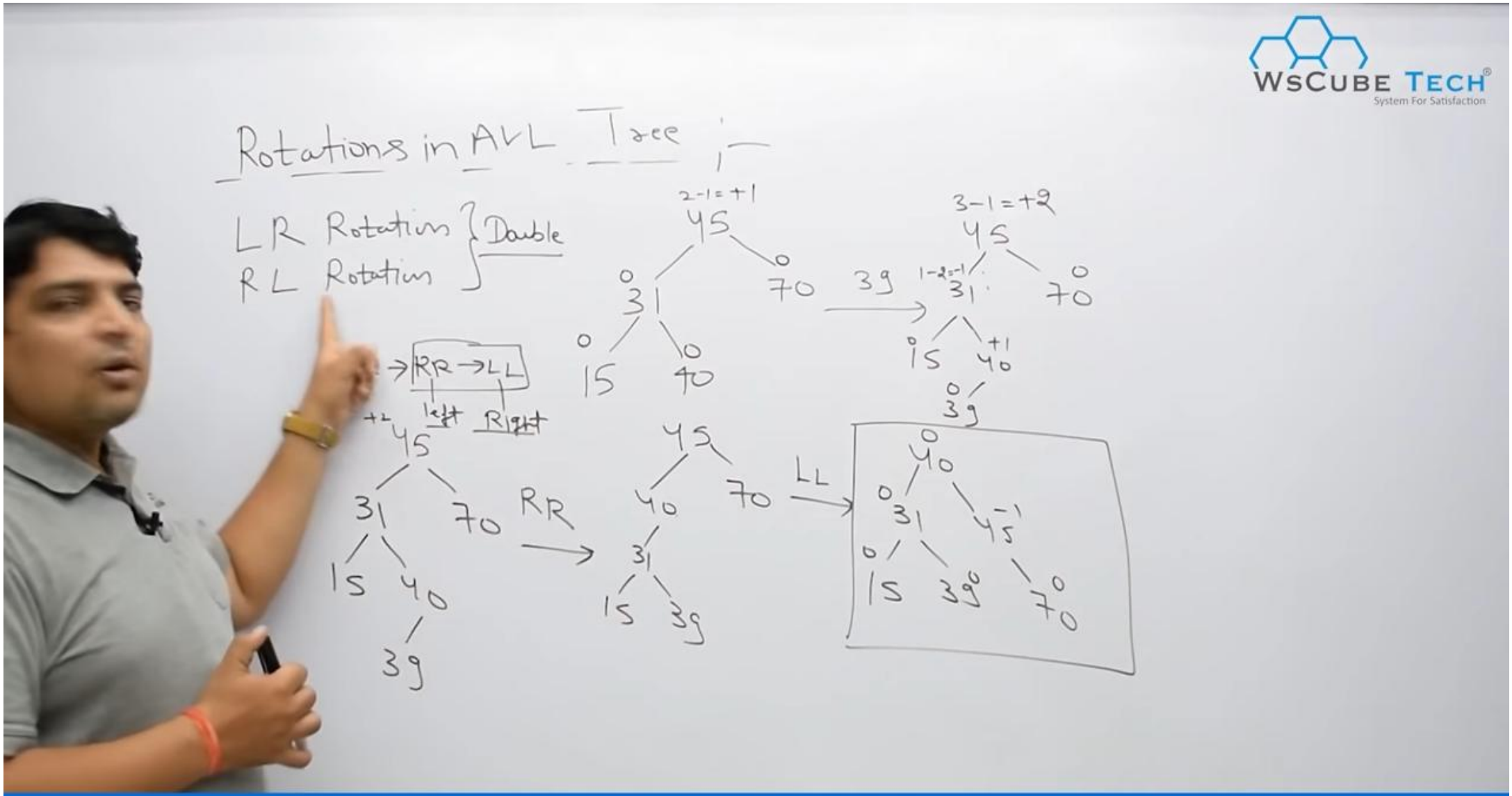
Data Structure & Algorithm

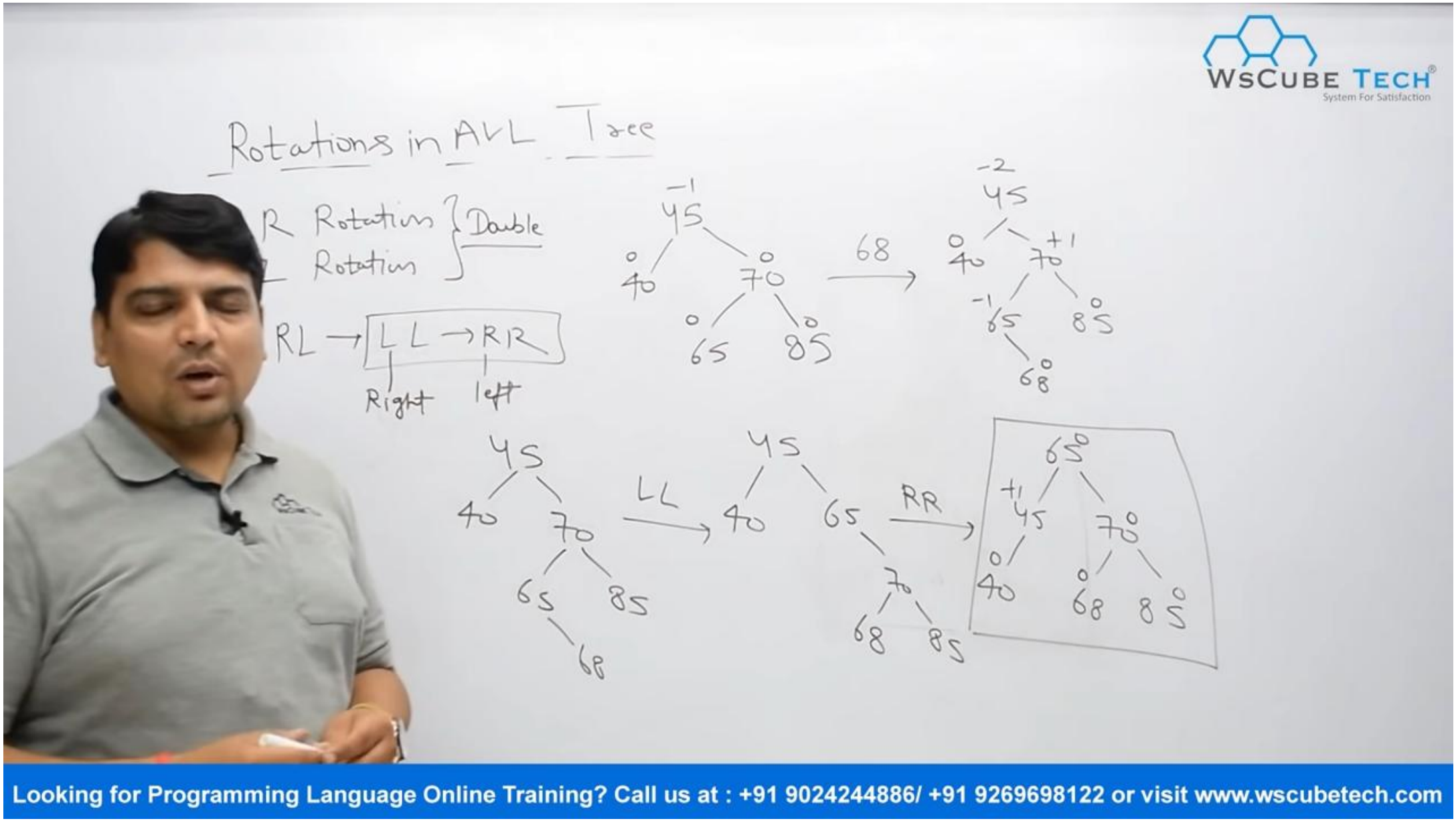
AVL Tree Rotation

Rotations in AVL Tree :-

LR Rotation } Double
RL Rotation }

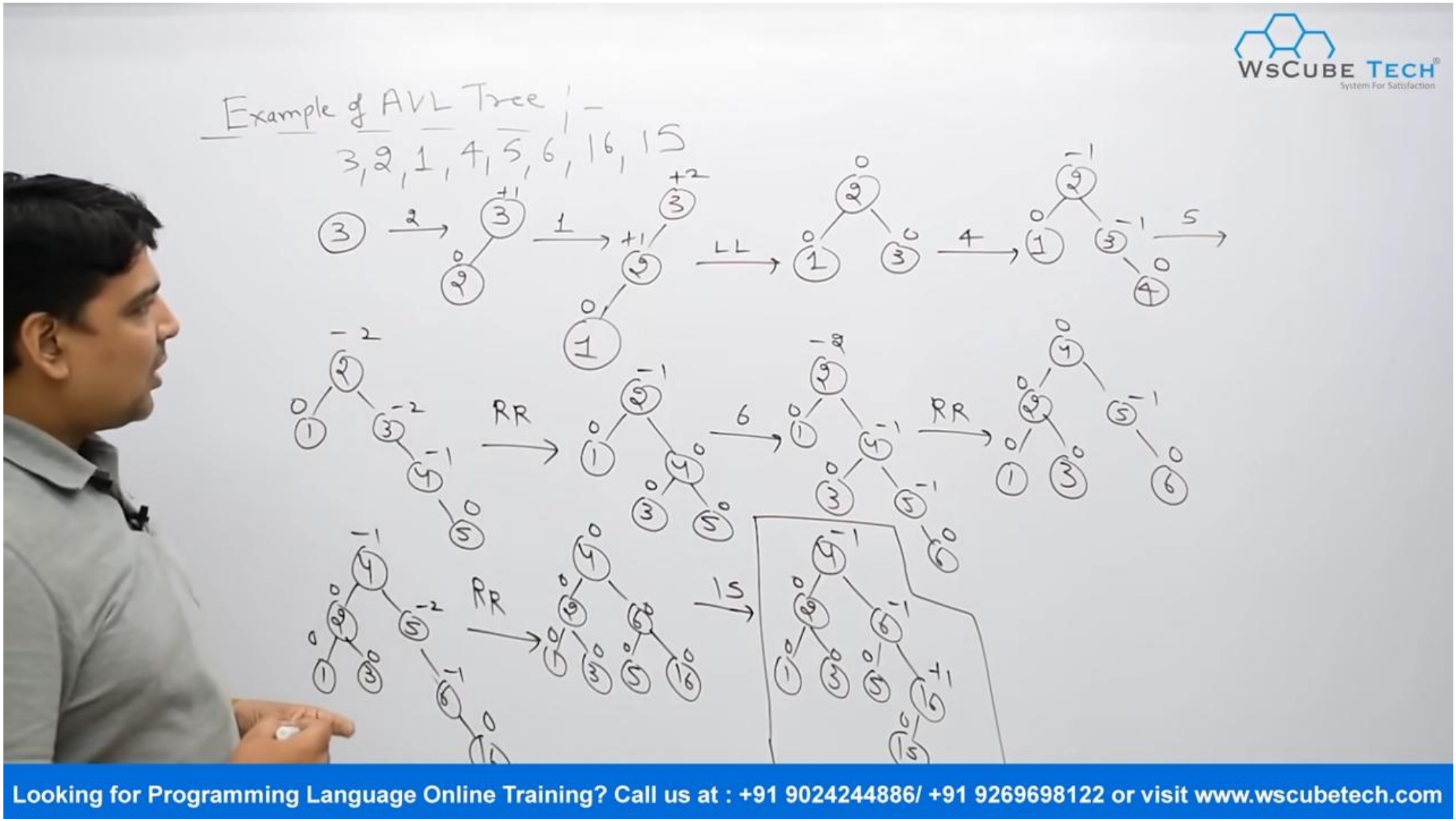



WsCube Tech
System For Satisfaction
Rotations in AVL Tree :-
LR Rotation
RL Rotation
Double
RR → LL
Left
Right
RR
LL
39
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com



Data Structure & Algorithm

AVL Tree Example

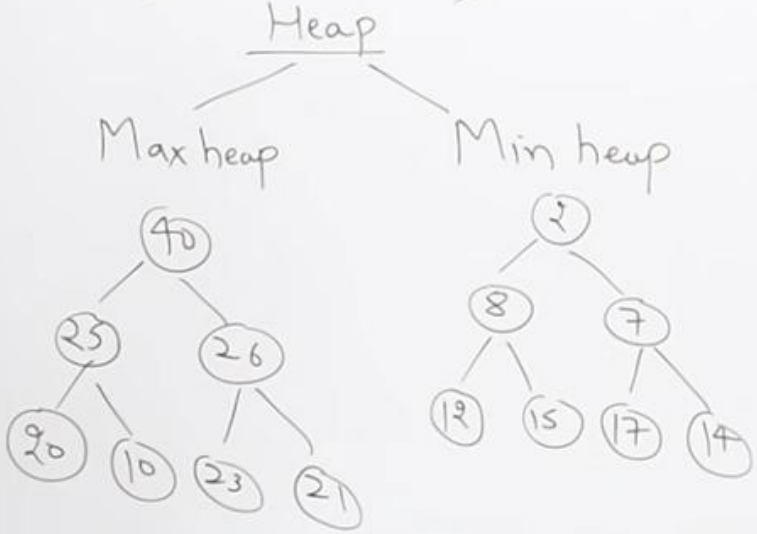


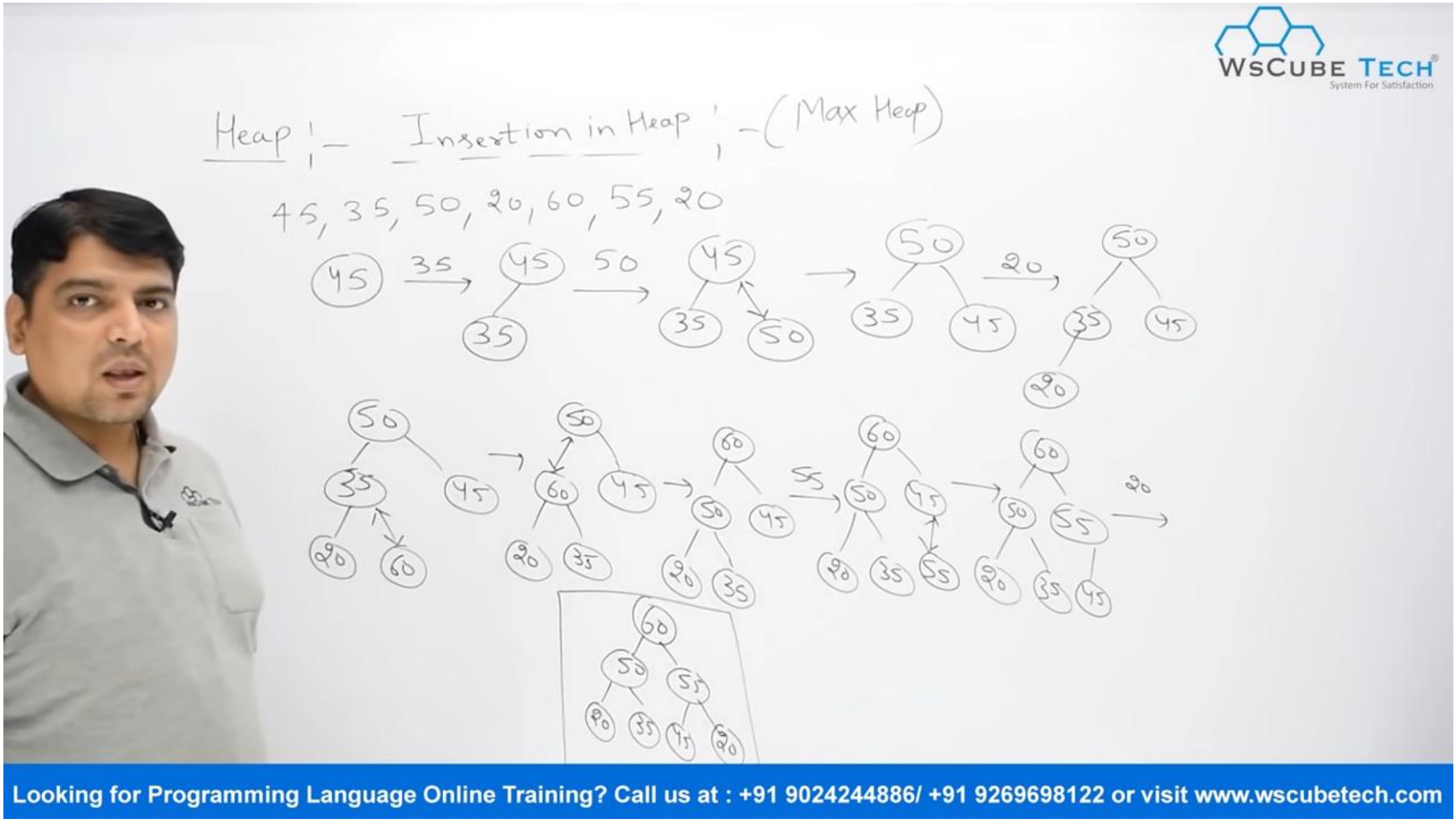
Data Structure & Algorithm

Insertion in Heap Tree

Heap :- It is a Complete Binary Tree

→ The value at Node N is greather than or equal to the value at each of the children of N (Max Heap)



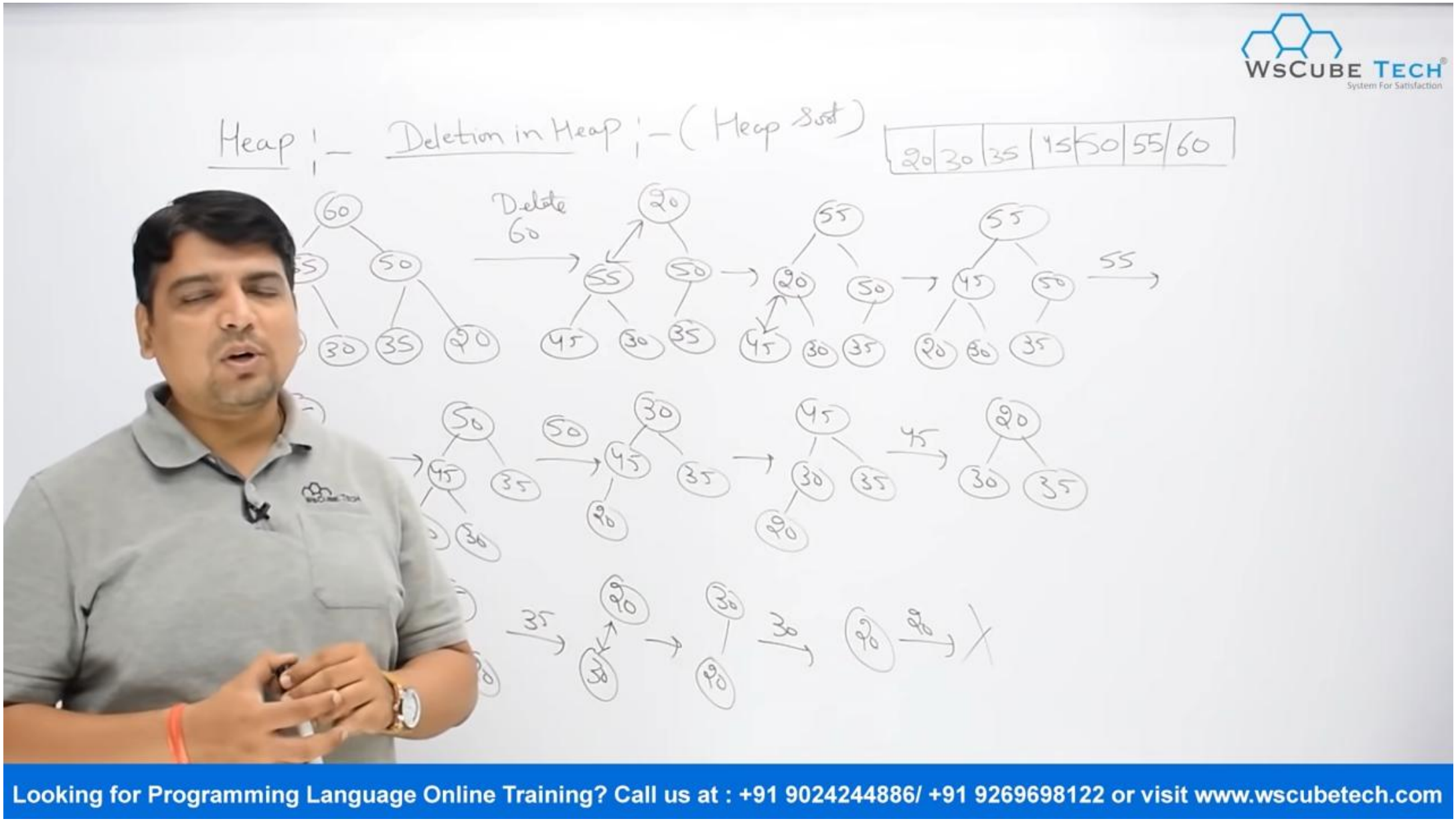

WSCUBE TECH
System For Satisfaction
Heap :- Insertion in Heap :- (Max Heap)
45, 35, 50, 20, 60, 55, 20
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

Data Structure & Algorithm

Deletion in Heap Tree

WsCube Tech
System For Satisfaction

Heap :- Deletion in Heap :-

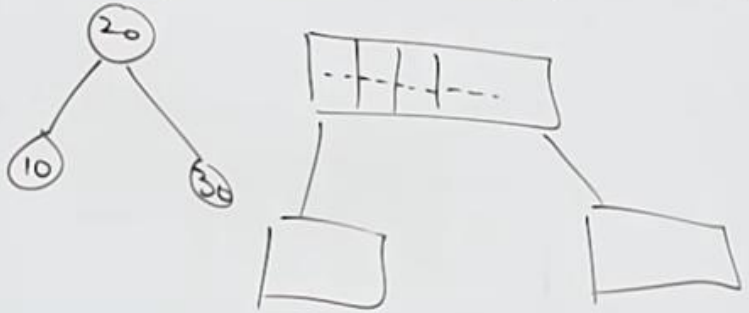


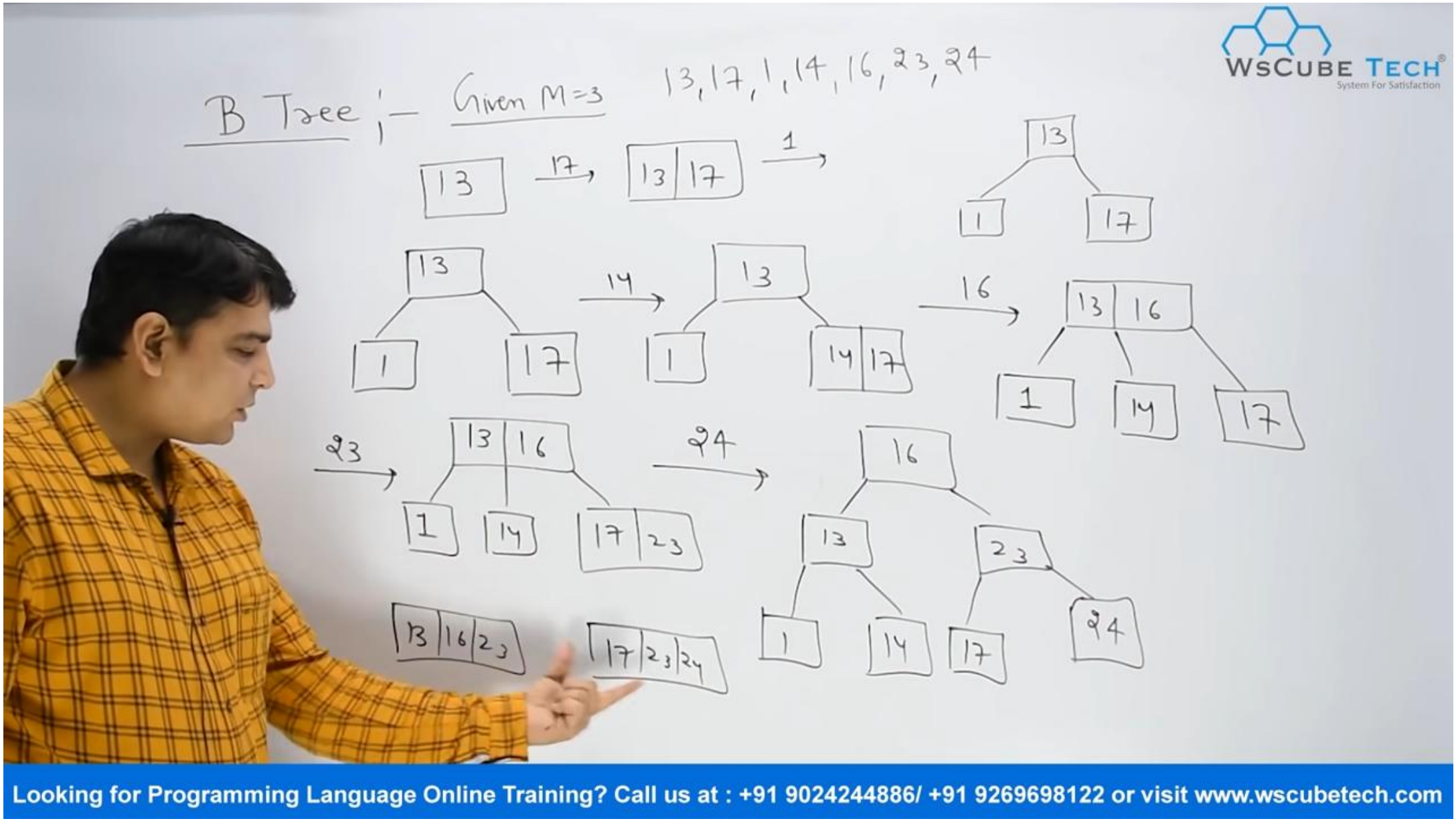
Data Structure & Algorithm

B Tree Insertion

B Tree :- B-Tree is a self-balanced Search Tree in which every node contains multiple keys (values) and has more than two children. (M-order)

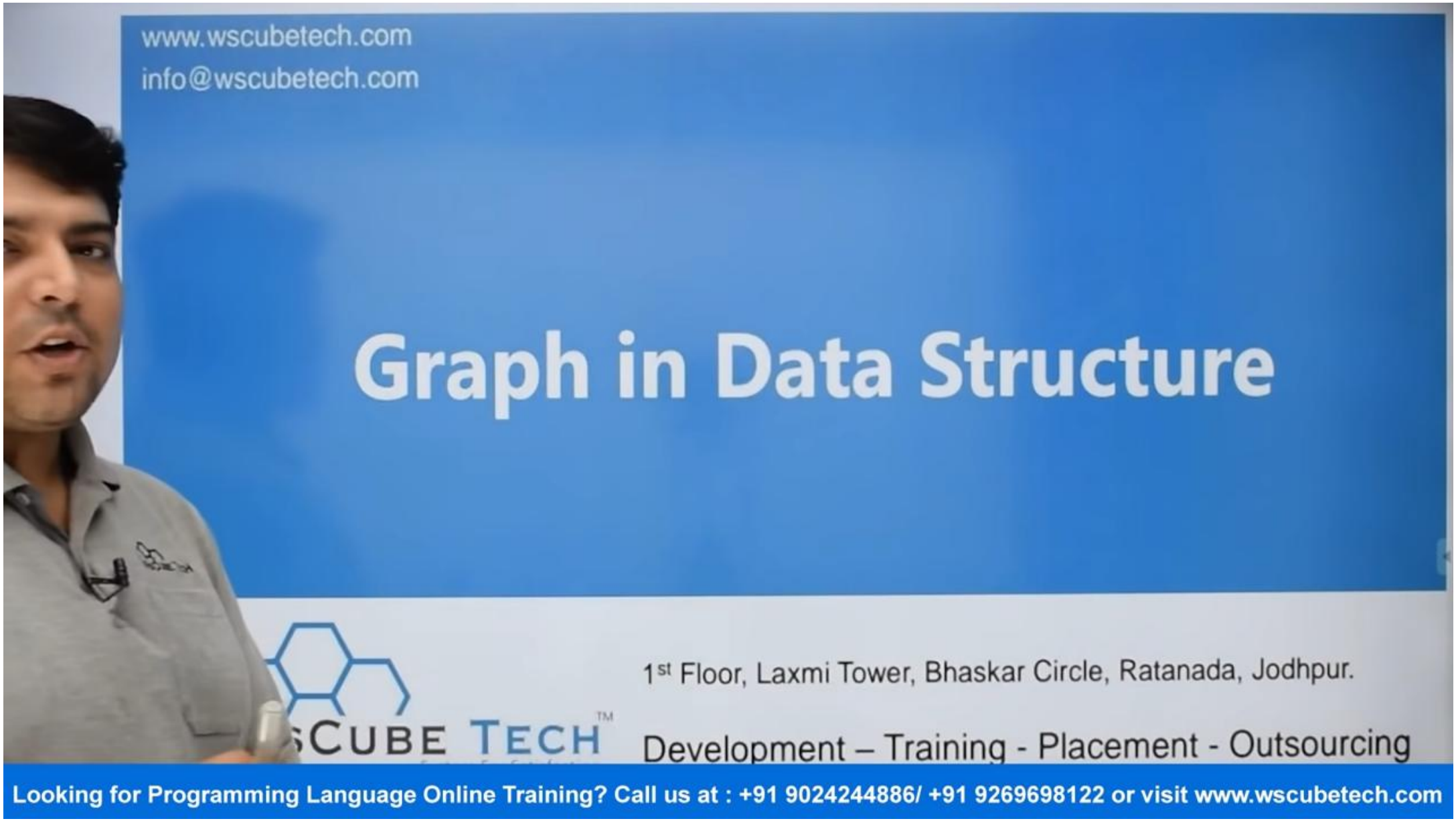
- → All Nodes (leaf) are at same level.
- → All nodes except root node have at least $\lceil m/2 \rceil - 1$ keys and max. m-1 keys.
- → No Node Contain more than (M-1) values
- → Keys are arranged in defined order within the Nodes.

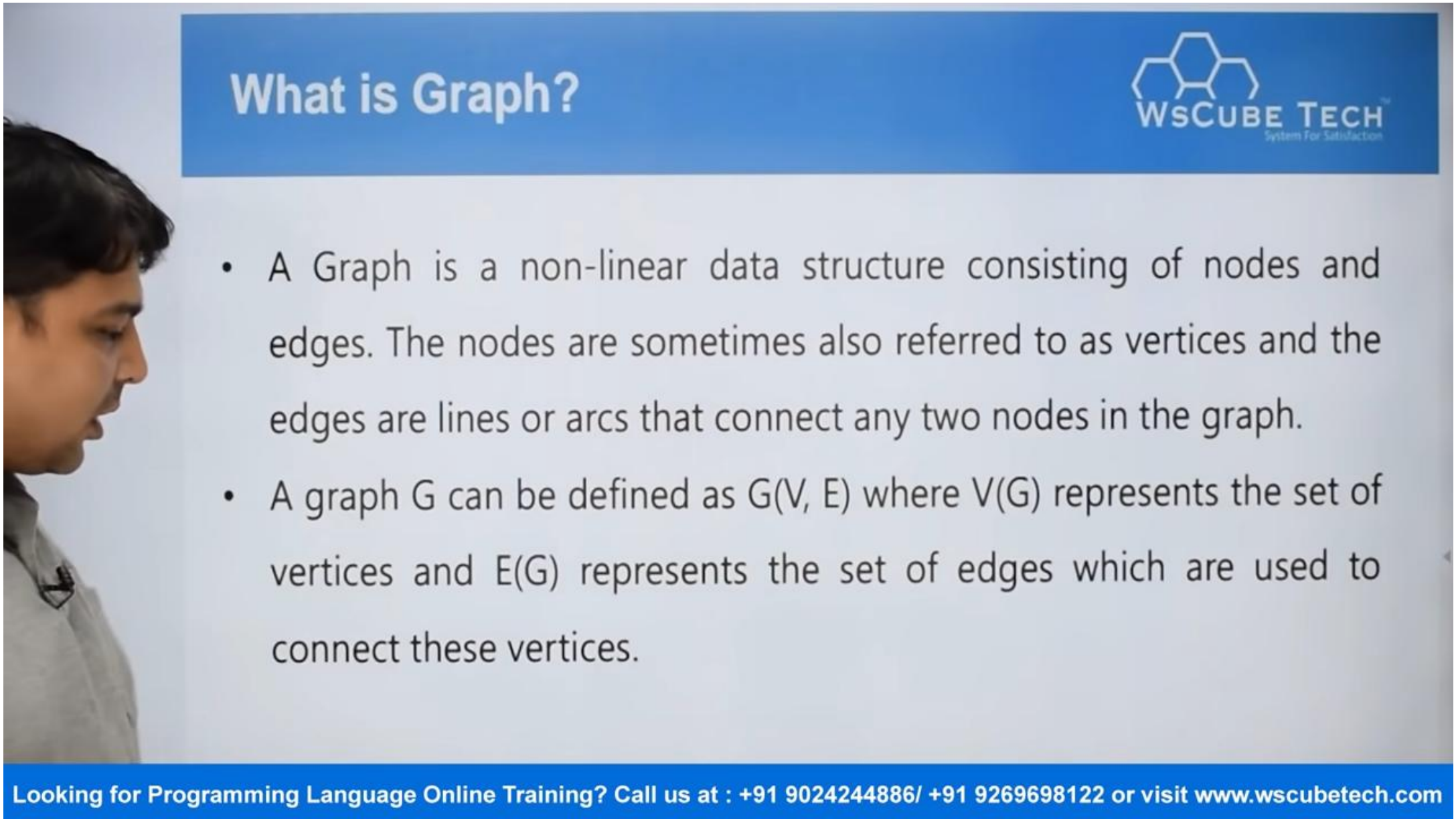


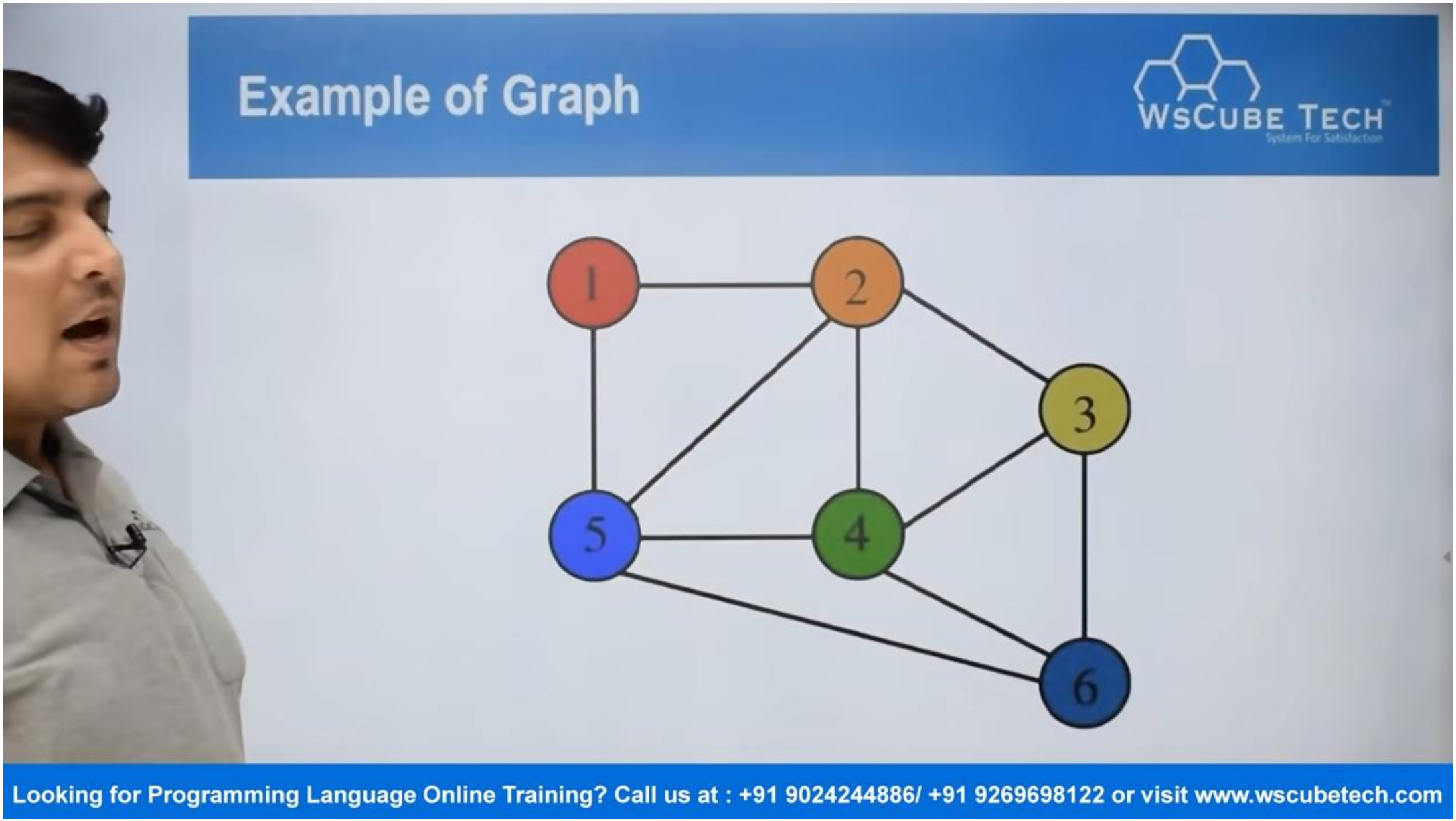


Data Structure & Algorithm

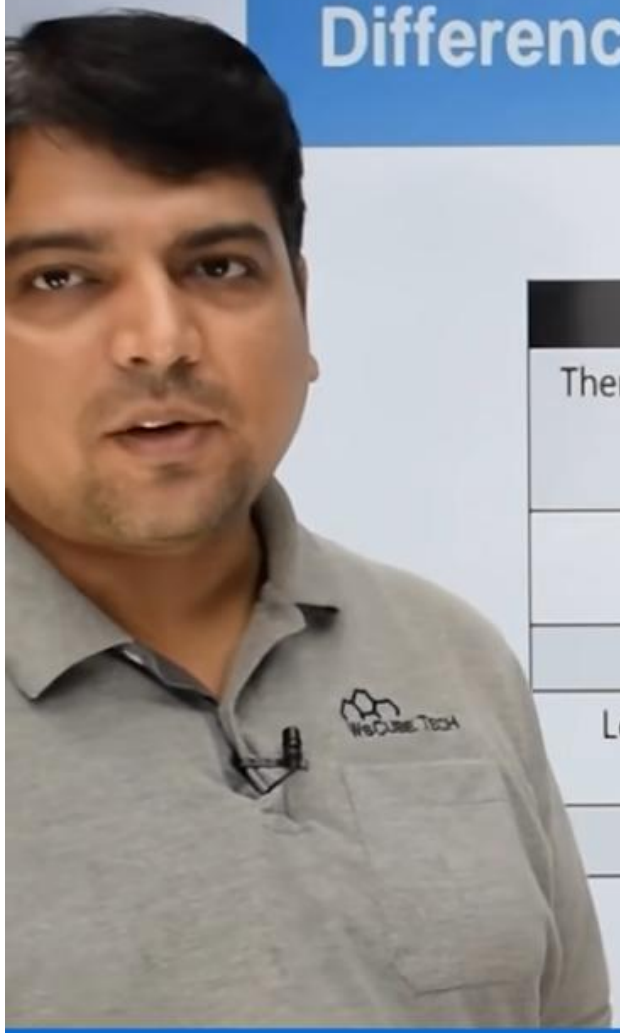
Introduction to Graph






Example of Graph
WSCUBE TECH
System For Satisfaction
1
2
3
4
5
6
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

Difference between Tree and Graph

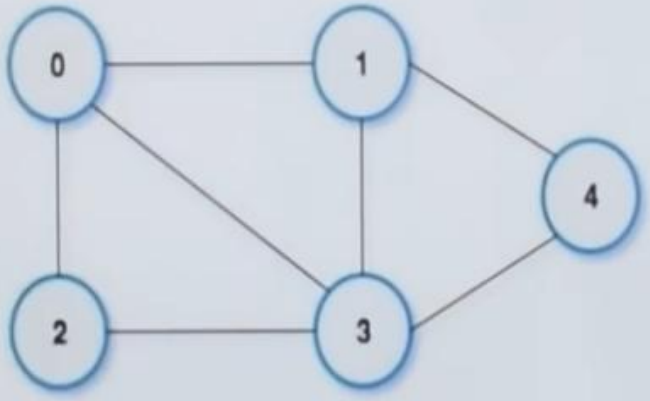


TREE	GRPAH
There is a unique node called root in trees.	All nodes called as root in graph.
There will not be any cycle.	A cycle can be formed.
All trees are graphs.	all graphs are not trees.
Less in complexity compared to graphs.	High complexity than trees due to loops.
Hierarchical	Network

Graph Terminology

WSCUBE TECH
System For Satisfaction

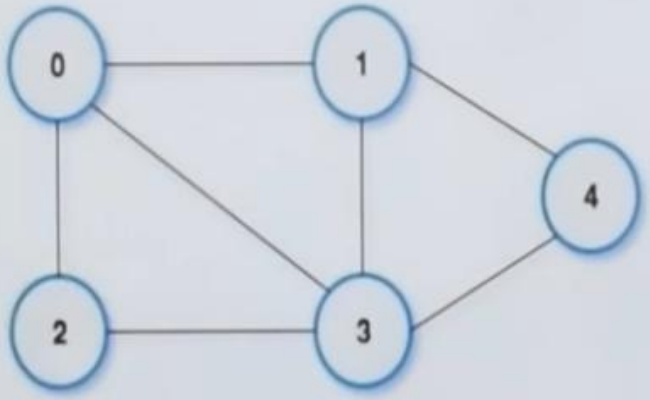
- **Undirected Graph:** A graph with only undirected edges is said to be undirected graph.
- **Directed Graph:** A graph with only directed edges is said to be directed graph.

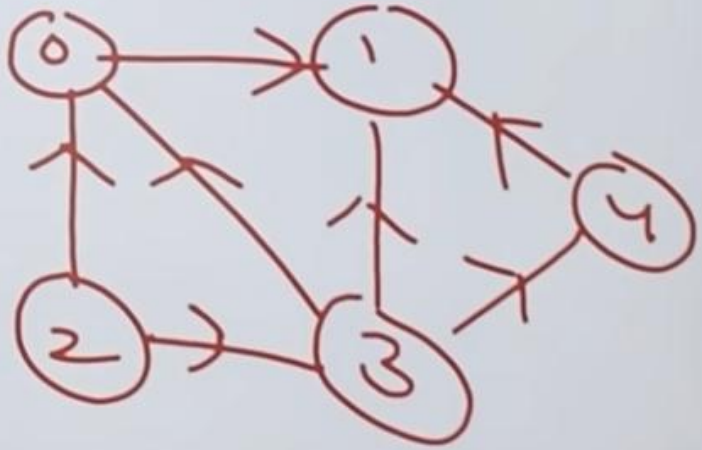


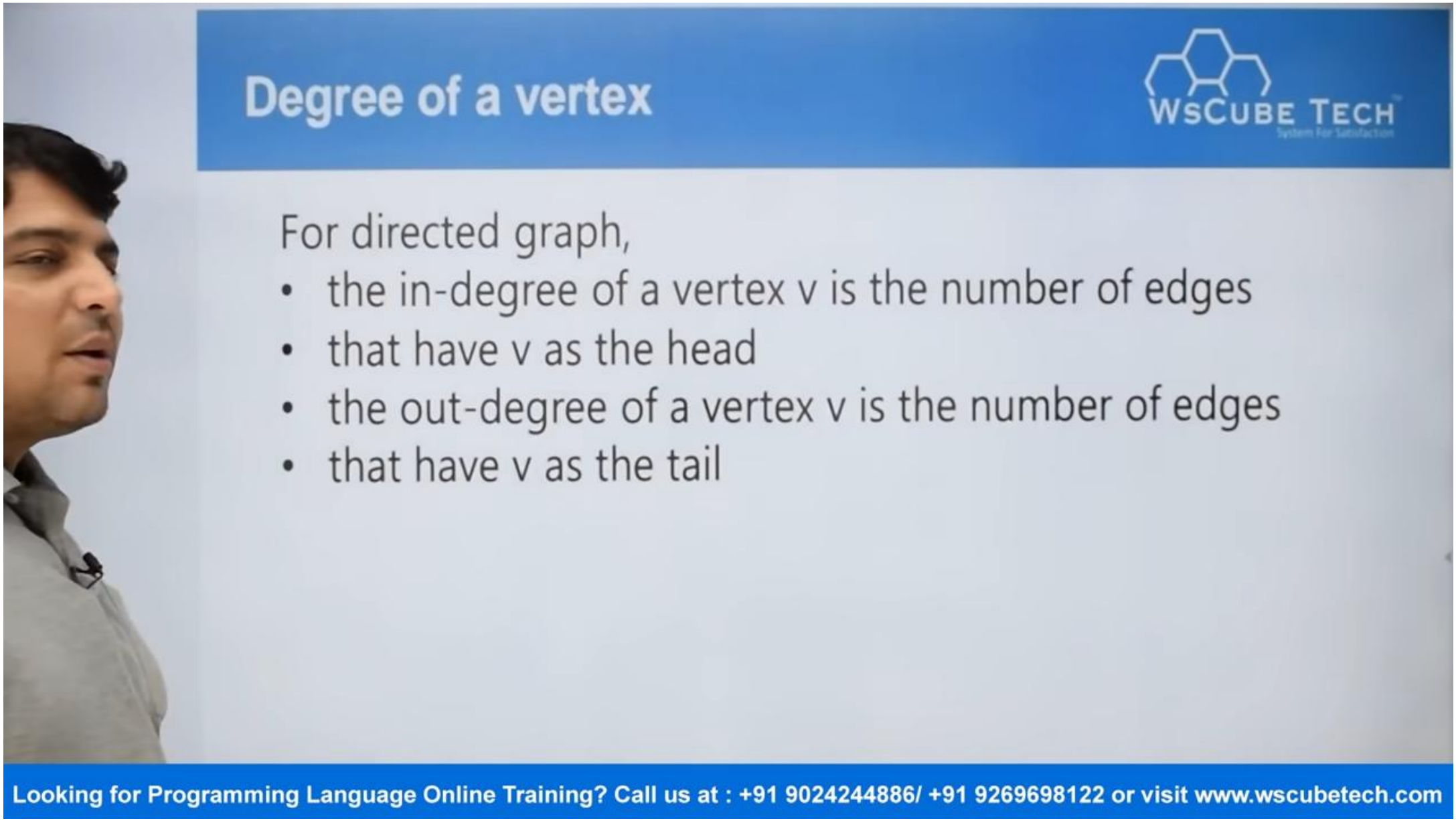
Graph Terminology

WSCUBE TECH
System For Satisfaction

- **Undirected Graph:** A graph with only undirected edges is said to be undirected graph.
- **Directed Graph:** A graph with only directed edges is said to be directed graph.







Degree of a vertex

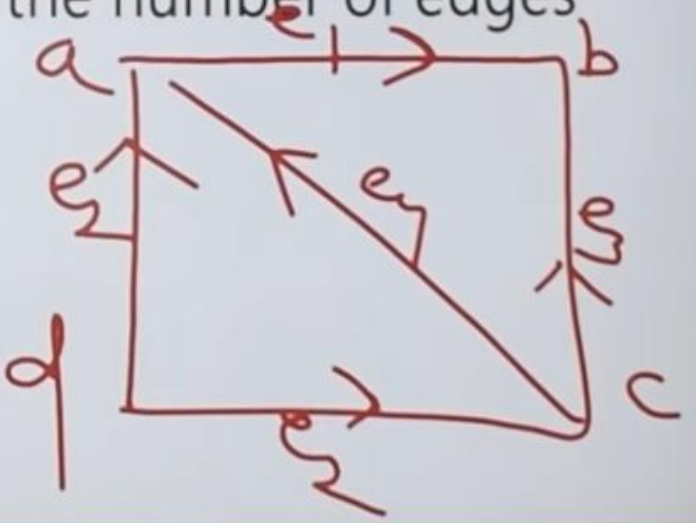
WSCUBE TECH
System For Satisfaction

For directed graph,

- the in-degree of a vertex v is the number of edges
- that have v as the head
- the out-degree of a vertex v is the number of edges
- that have v as the tail

deg−(a) = 2

deg+(a) = 1



Simple Graph

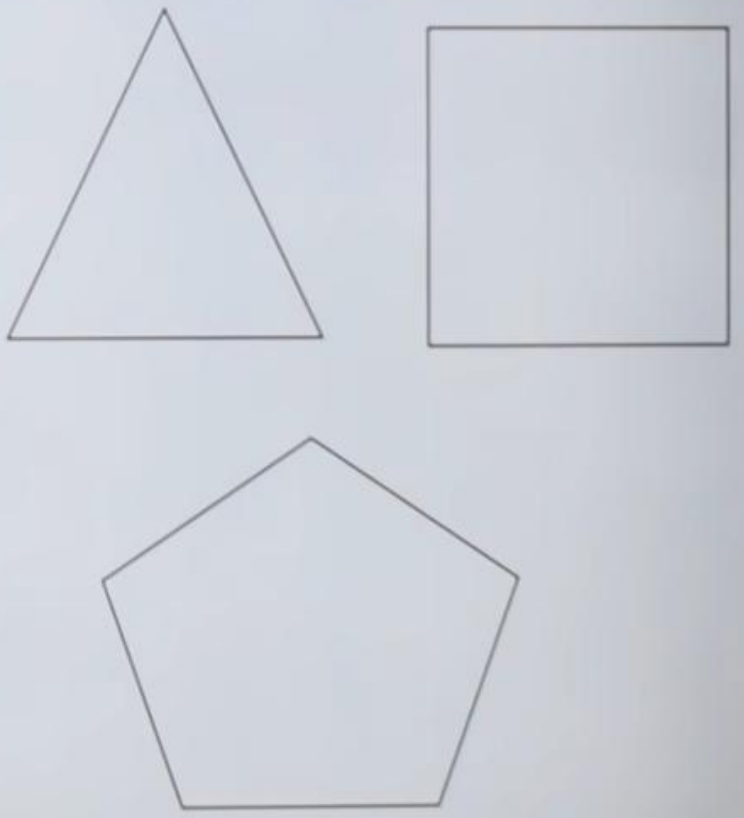
WsCube Tech
System For Satisfaction

A graph with no loops and no parallel edges is called a simple graph.

Cycle Graph

WsCube Tech
System For Satisfaction

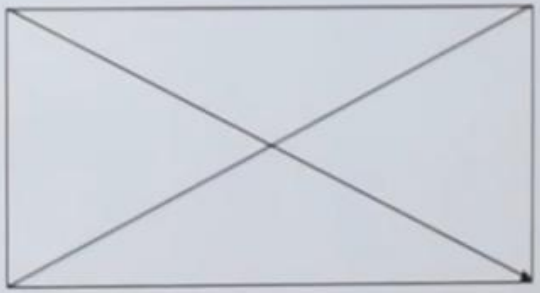
A simple graph with 'n' vertices (n >= 3) and 'n' edges is called a cycle graph if all its edges form a cycle of length 'n'.
If the degree of each vertex in the graph is two, then it is called a Cycle Graph. It is denoted by Cn.



Complete Graph

WSCUBE TECH
System For Satisfaction

A vertex is connected to all other vertices in a graph, then it is called a complete graph.



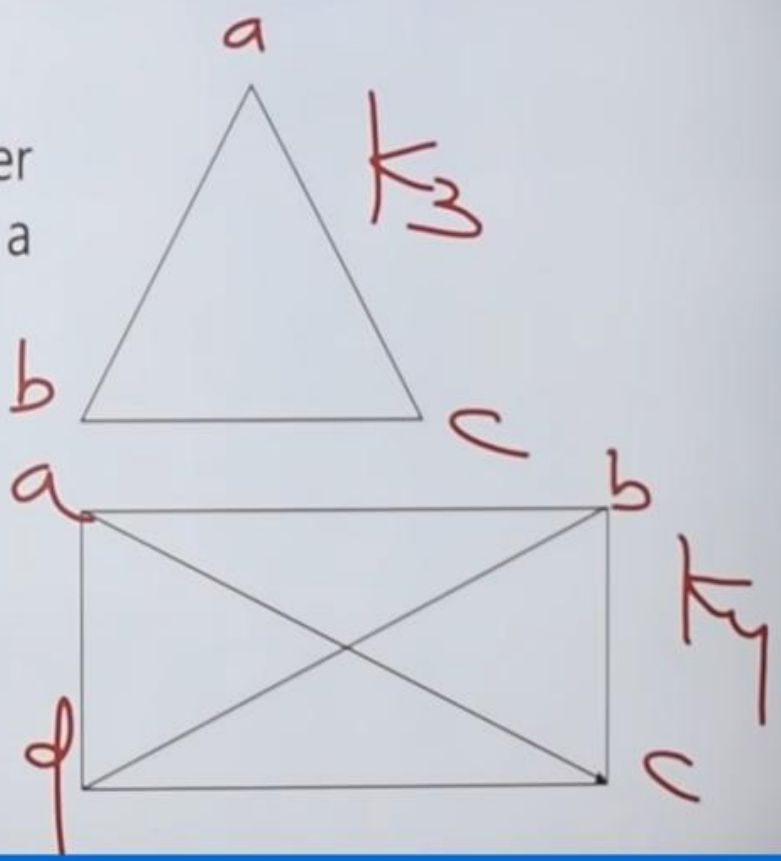
Complete Graph

WSCUBE TECH
System For Satisfaction

K_8 K_{10}

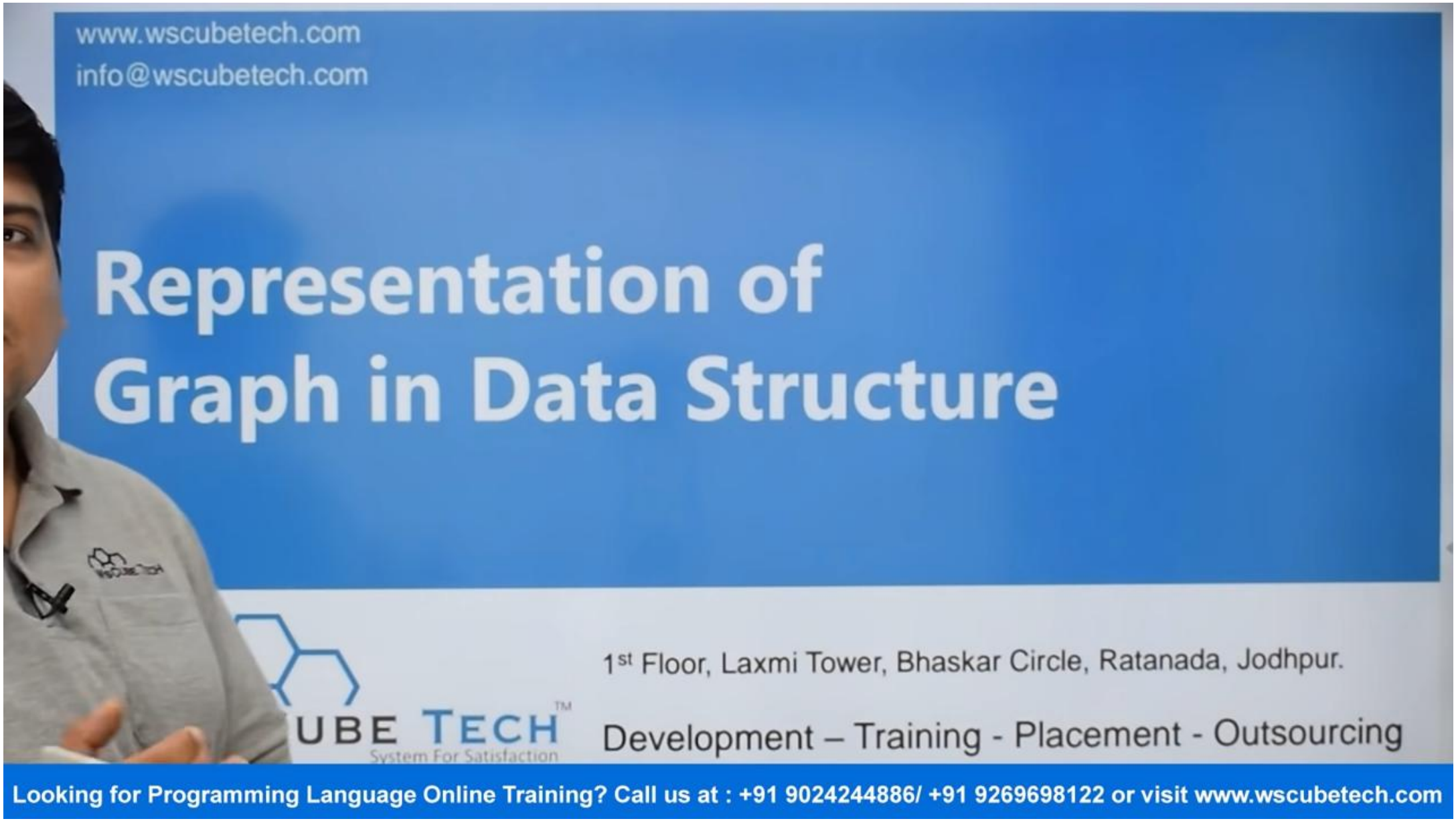
A vertex is connected to all other vertices in a graph, then it is called a complete graph.

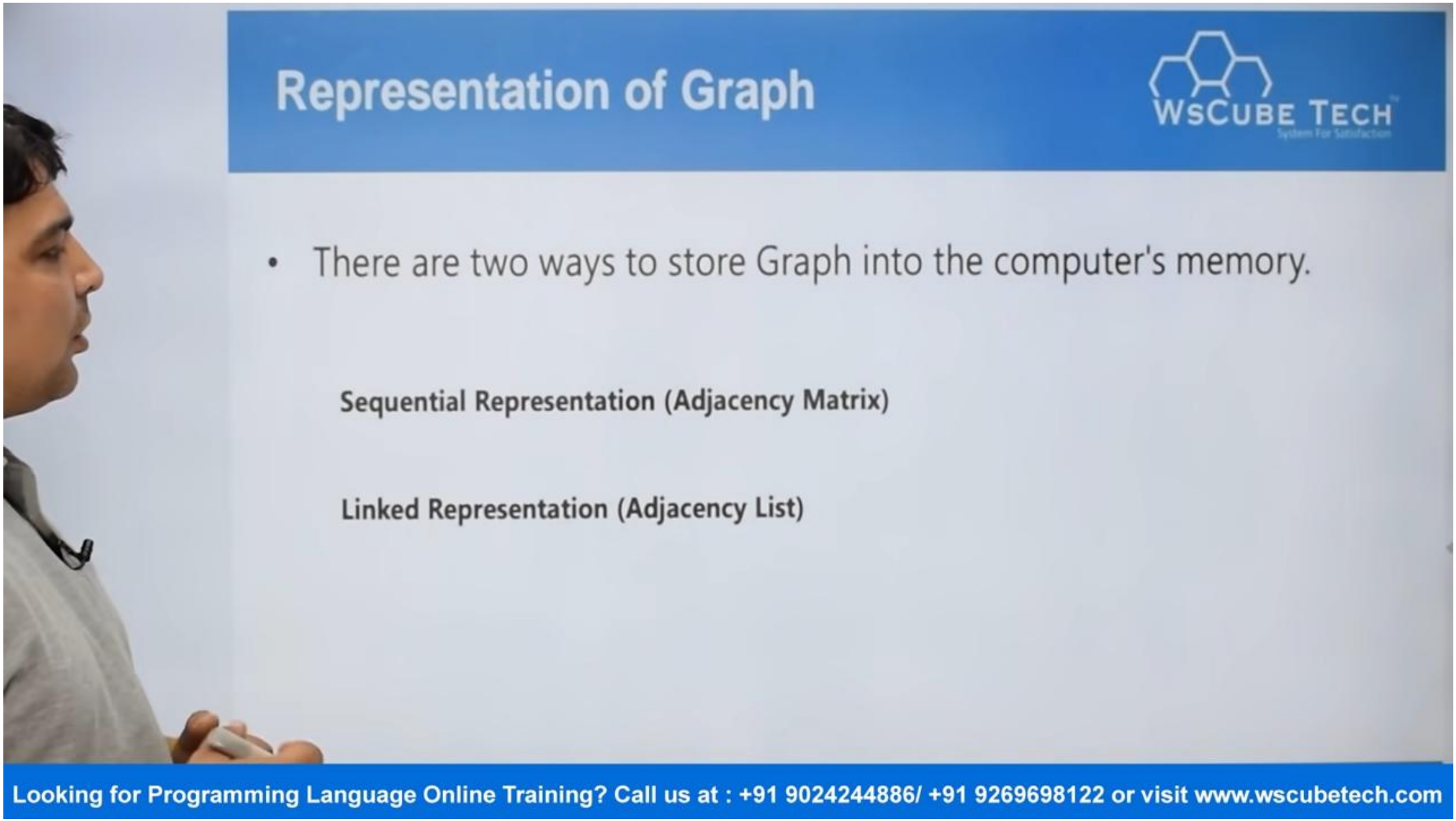
$$K_n = \frac{n(n-1)}{2}$$



Data Structure & Algorithm

Representation of Graph

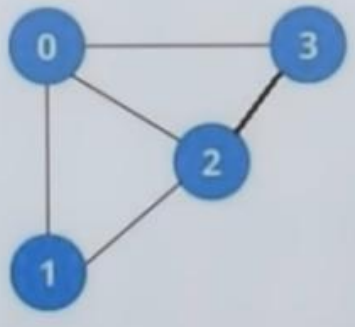




Sequential Representation

WsCube Tech
System For Satisfaction

An adjacency matrix is 2D array of V x V vertices. Each row and column represent a vertex. If the value of any element a[i][j] is 1, it represents that there is an edge connecting vertex i and vertex j.

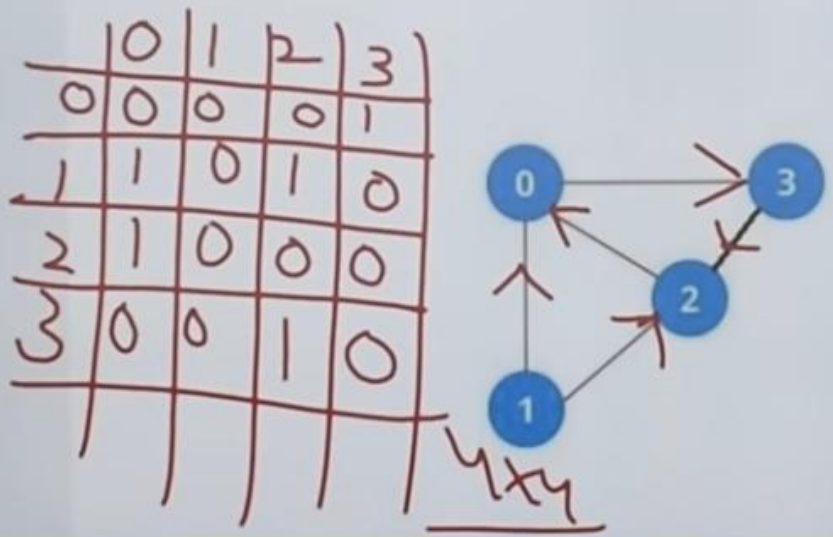


	0	1	2	3
0	0	1	1	1
1	1	0	1	0
2	1	1	0	1
3	1	0	1	0

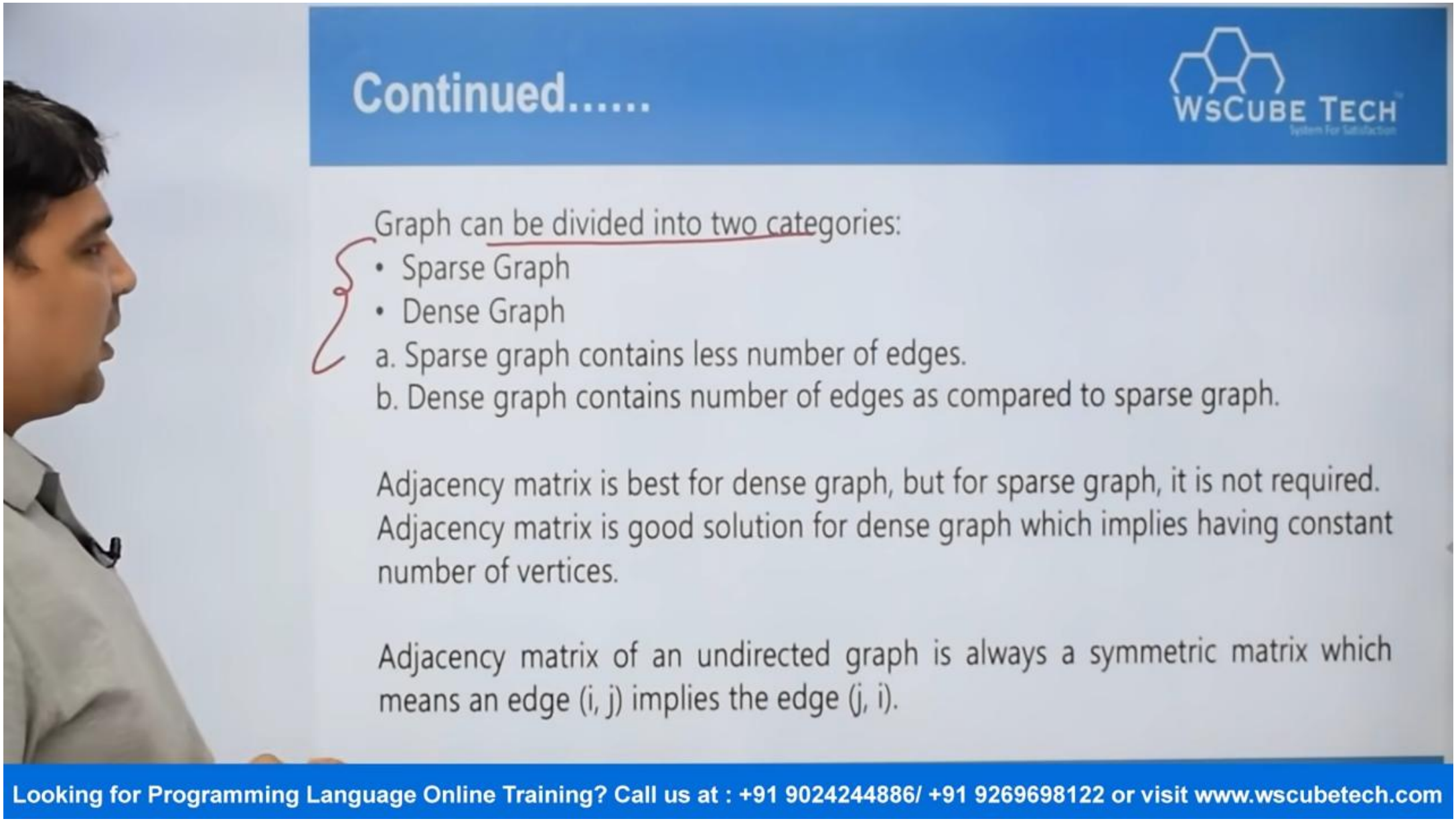
Sequential Representation

WsCube Tech
System For Satisfaction

An adjacency matrix is 2D array of V x V vertices. Each row and column represent a vertex. If the value of any element a[i][j] is 1, it represents that there is an edge connecting vertex i and vertex j.



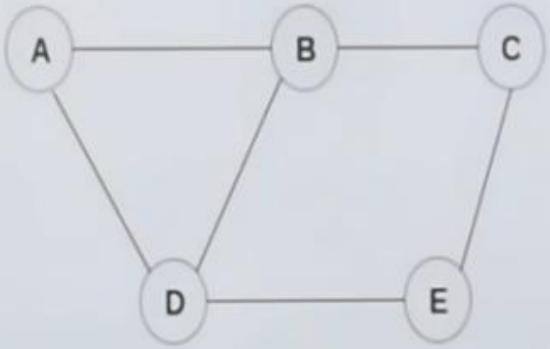
	0	1	2	3
0	0	1	1	1
1	1	0	1	0
2	1	1	0	1
3	1	0	1	0



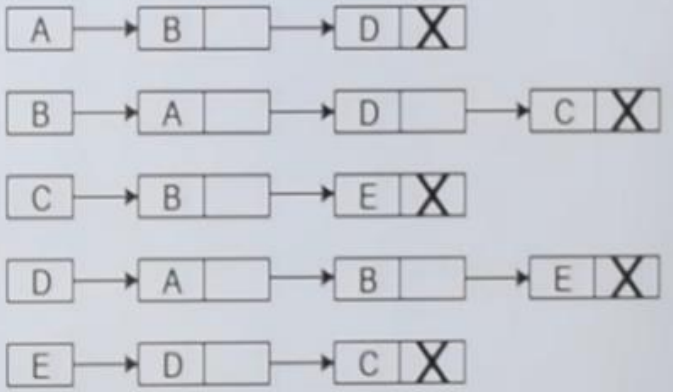
Linked Representation

WsCube Tech

An adjacency list is maintained for each node present in the graph which stores the node value and a pointer to the next adjacent node to the respective node.



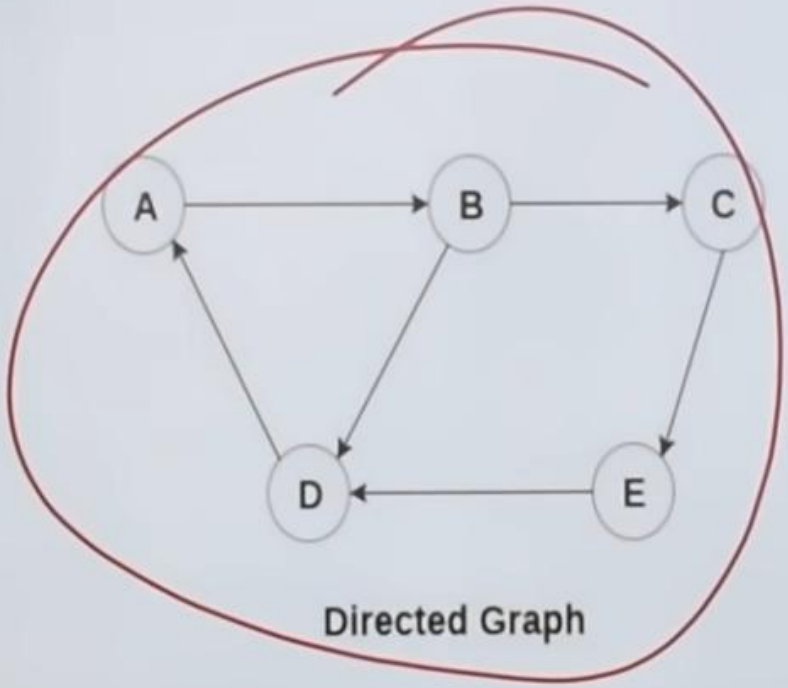
Undirected Graph



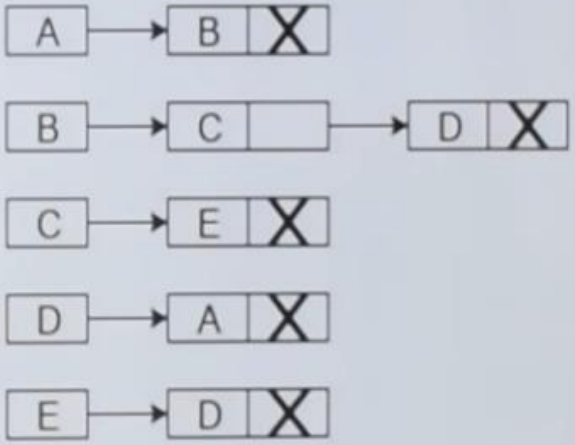
Adjacency List

Continued.....

WsCube Tech
System For Satisfaction



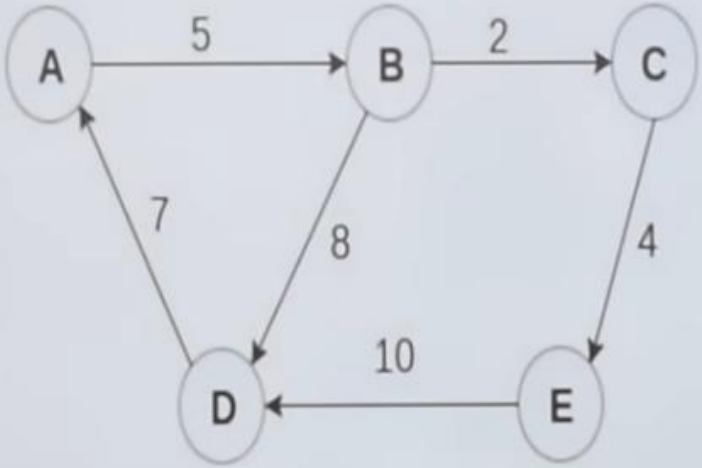
Directed Graph



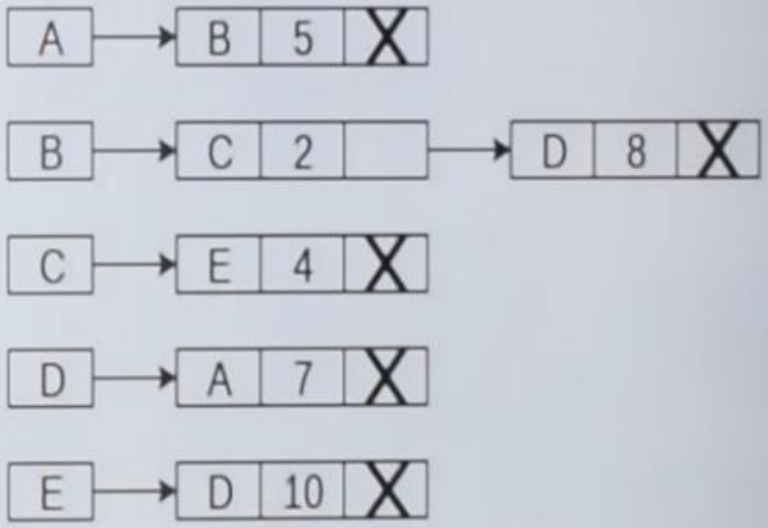
Adjacency List

Continued.....

WSCUBE TECH
System For Satisfaction



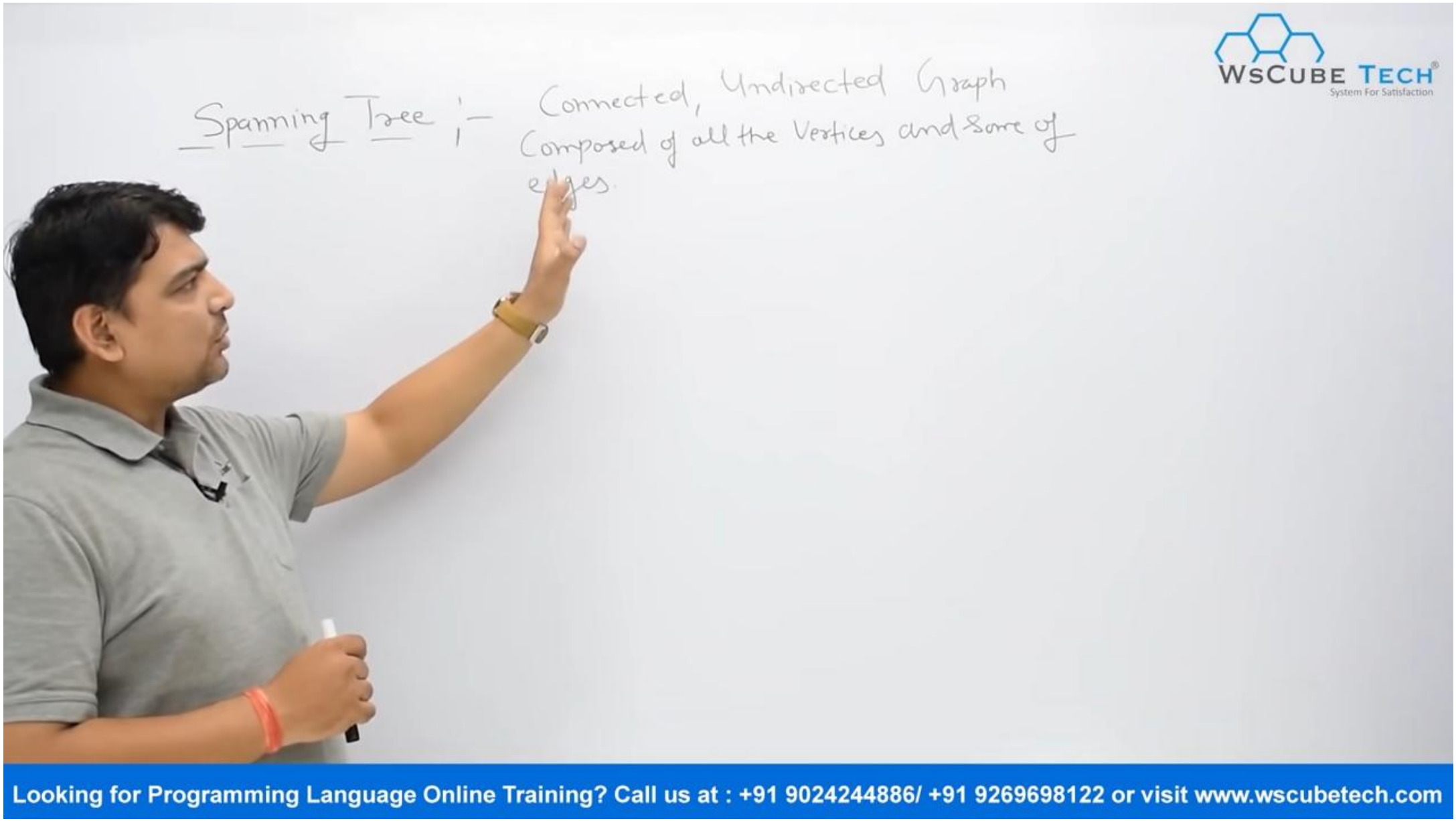
Weighted Directed Graph



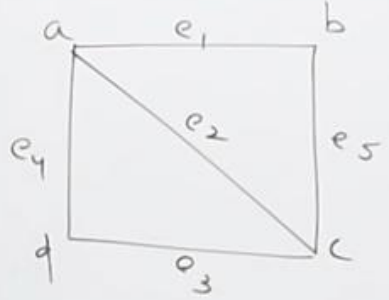
Adjacency List

Data Structure & Algorithm

Spanning Tree

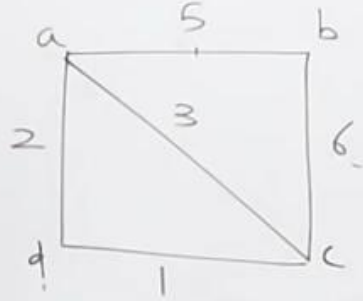


Spanning Tree :- Connected, Undirected Graph Composed of all the Vertices and Some of edges.



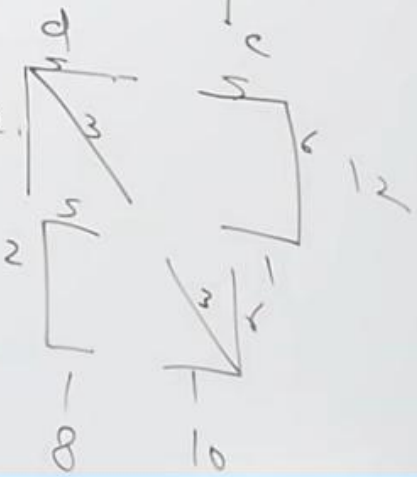
for n vertices, to construct the Spanning Tree, we Required n vertices and (n-1) edges

Spanning Tree :- Connected, Undirected Graph Composed of all the Vertices and Some of edges.



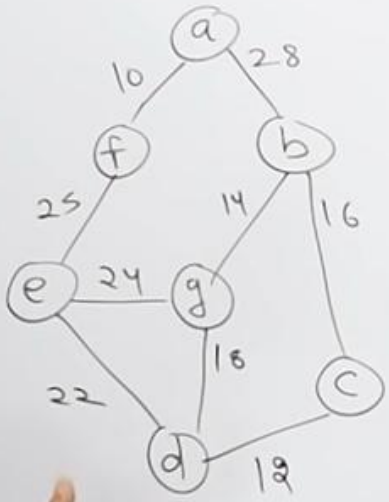
n = 4 | (n-1) = (3)

for n vertices, to constitute the Spanning Tree, we Required 102.
n vertices and (n-1) edges
No loop or cycle Exist.



WSCUBE TECH
System For Satisfaction

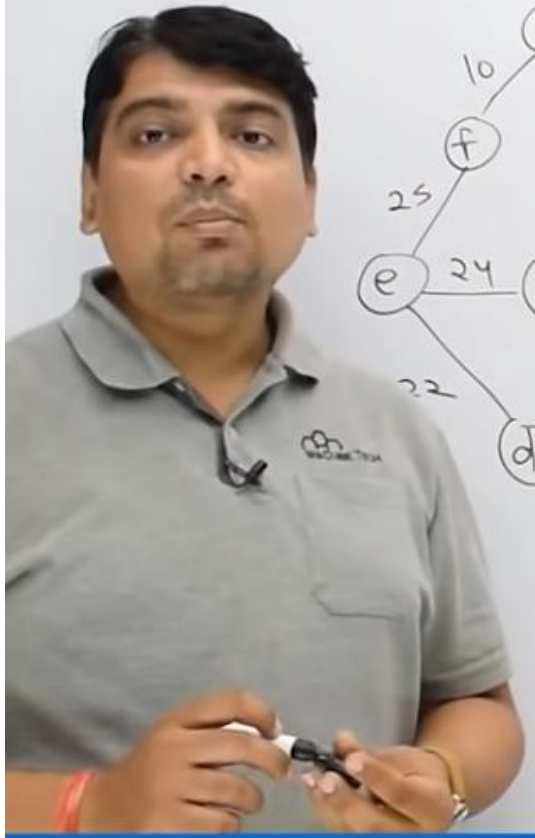
Spanning Tree :- Kruskal's Algorithms

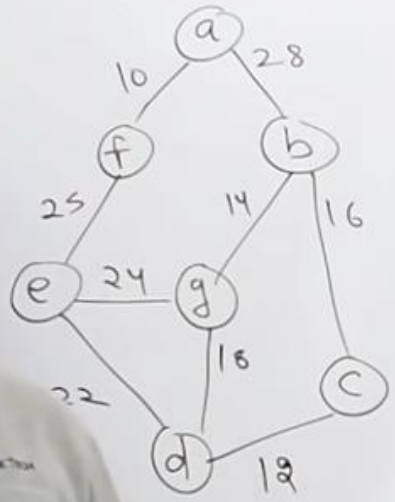


10	12	14	16	18	22	24	25	28
(a,f)	(c,d)	(b,g)	(b,c)	(d,g)	(d,e)	(e,g)	(e,f)	(a,b)

WSCUBE TECH
System For Satisfaction

Spanning Tree :- Kruskal's Algorithms

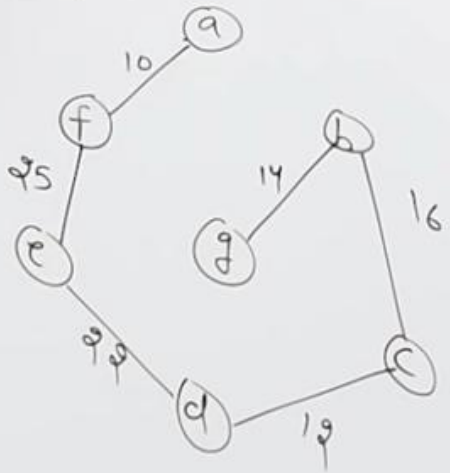




n = 7

(n-1) = 6

✓	✓	✓	✓	X	✓	X	✓	
10	12	14	16	18	22	24	25	28
(a,f)	(c,d)	(b,g)	(b,c)	(d,g)	(d,e)	(e,g)	(e,f)	(a,b)

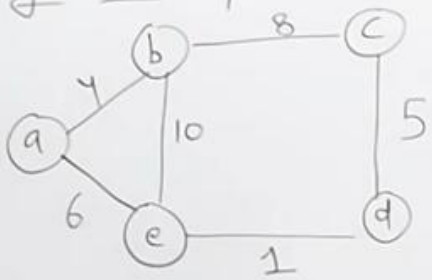


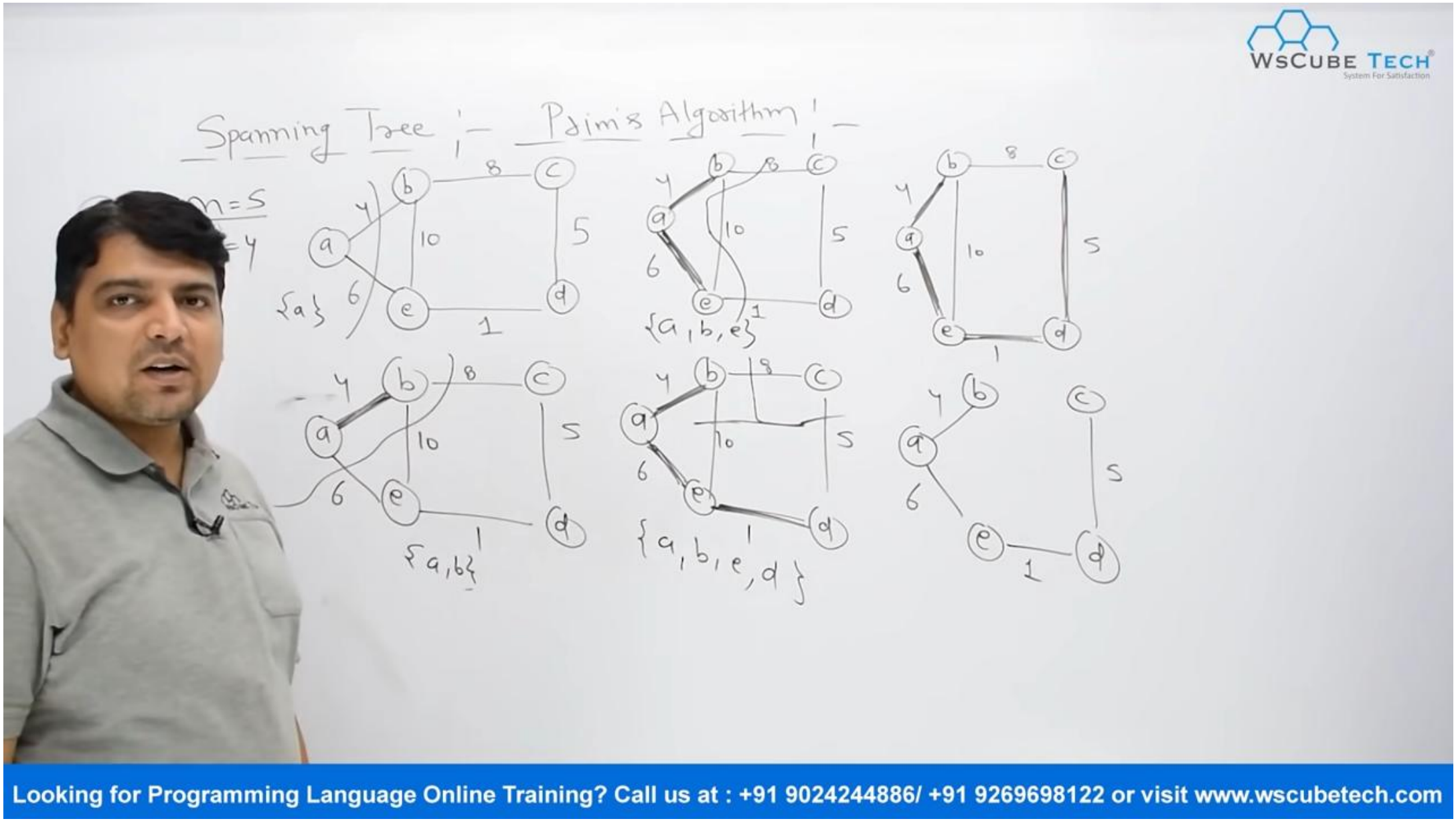
Data Structure & Algorithm

Prim's Algorithm

WsCube Tech
System For Satisfaction

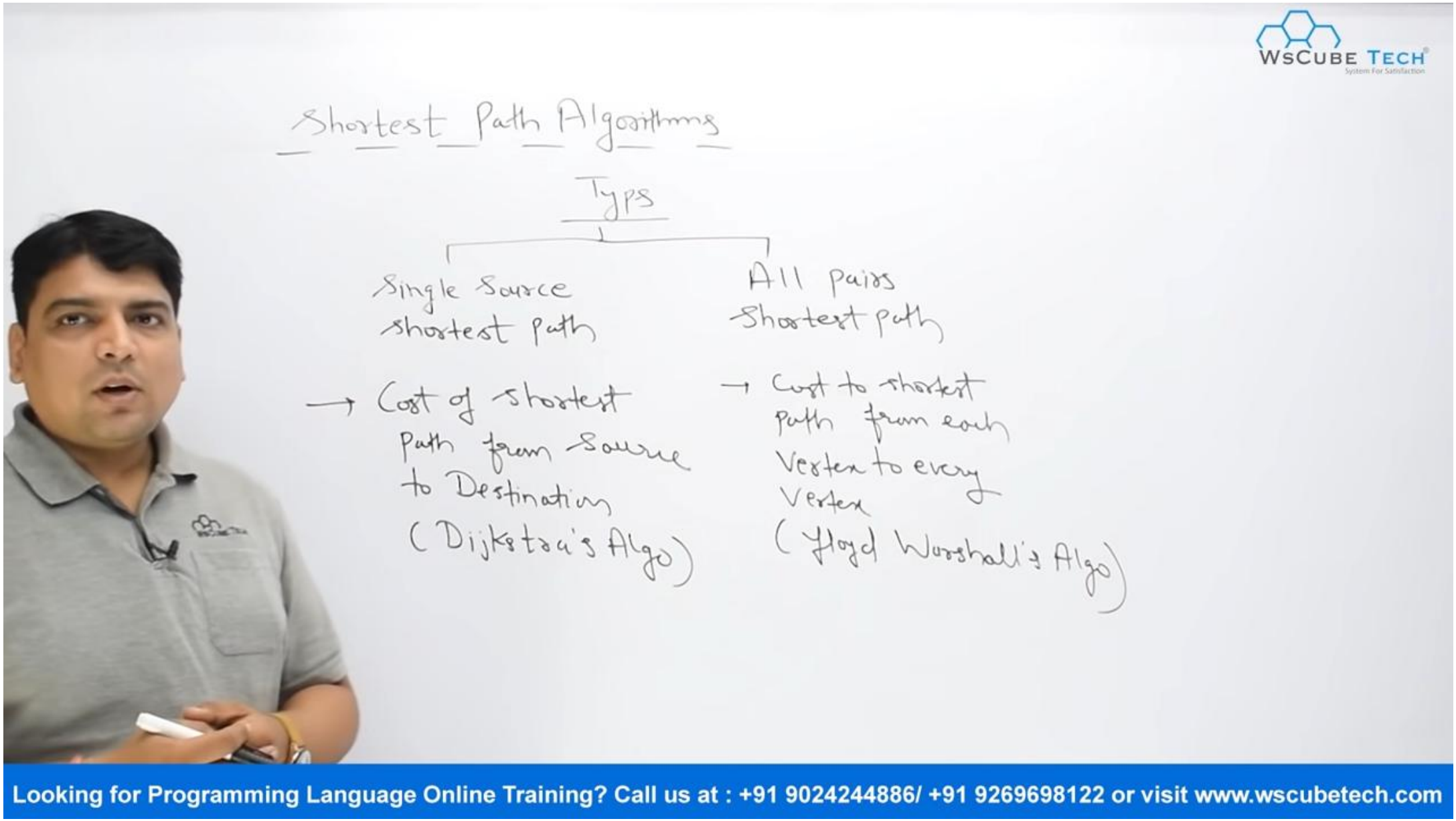
Spanning Tree :- Prim's Algorithm :-




WsCube Tech
System For Satisfaction
Spanning Tree :- Prim's Algorithm :-
{a}
{a, b, e}
{a, b}
{a, b, e, d}
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

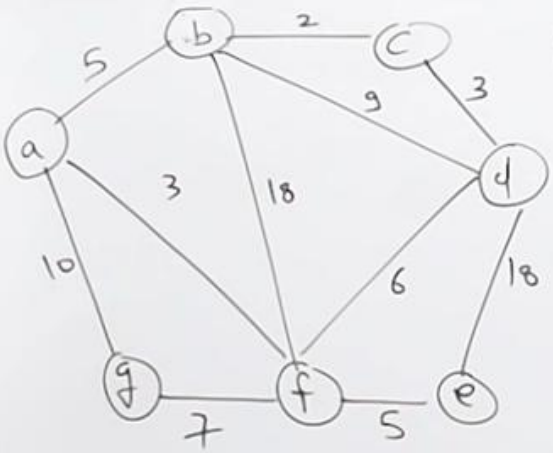
Data Structure & Algorithm

Shortest Path Algorithms in DSA



WsCube Tech
System For Satisfaction

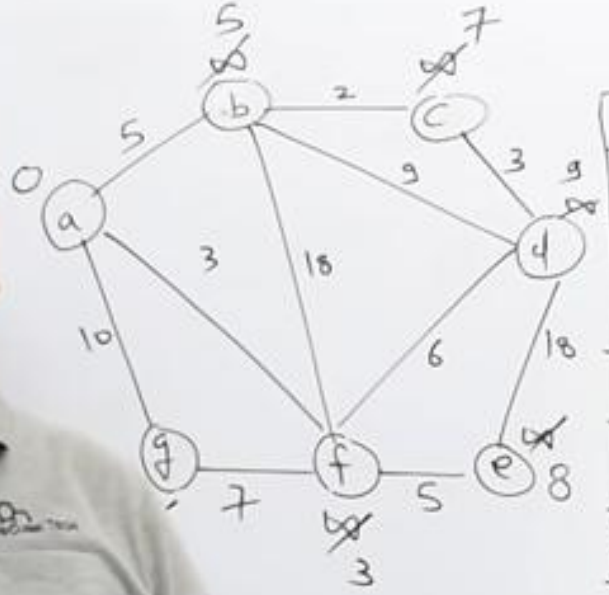
Single Source Shortest Path Algorithm !—(Dijkstra's)



	a	b	c	d	e	f	g
{a}	0	∞	∞	∞	∞	∞	∞

WsCube Tech

Single Source Shortest Path Algorithm !—(Dijkstra's)



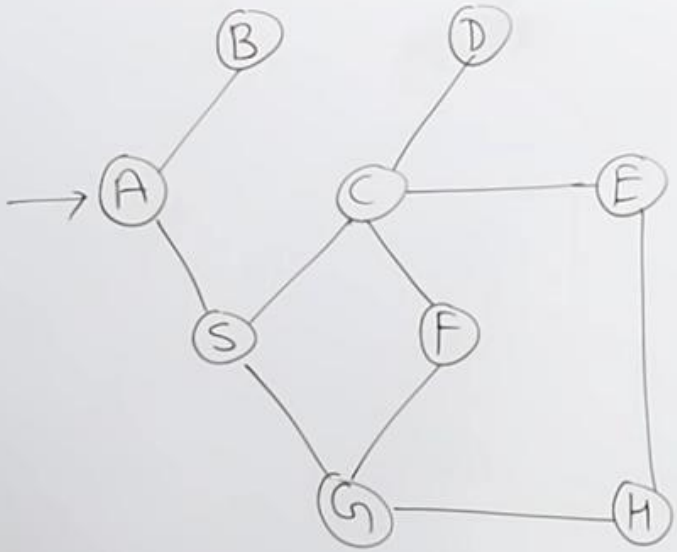
	a	b	c	d	e	f	g
{a}	0	∞	∞	∞	∞	∞	∞
{a,f}	0	5	∞	∞	∞	3	10
{a,f,b}	0	5	∞	9	8	3	10
{a,f,b,c}	0	5	7	9	8	3	10
{a,f,b,c,e}	0	5	7	9	8	3	10
{a,f,b,c,e,d}	0	5	7	9	8	3	10
{a,f,b,c,e,d,g}	0	5	7	9	8	3	10
	0	5	7	9	8	3	10

Data Structure & Algorithm

Graph Traversal Algorithm

WsCube Tech
System For Satisfaction

Graph Traversal Algorithms :- Breadth First Search (BFS)

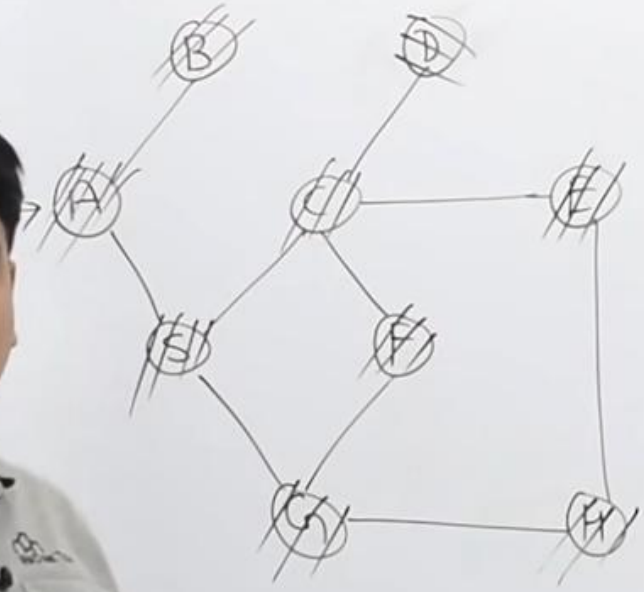


Queue

Output

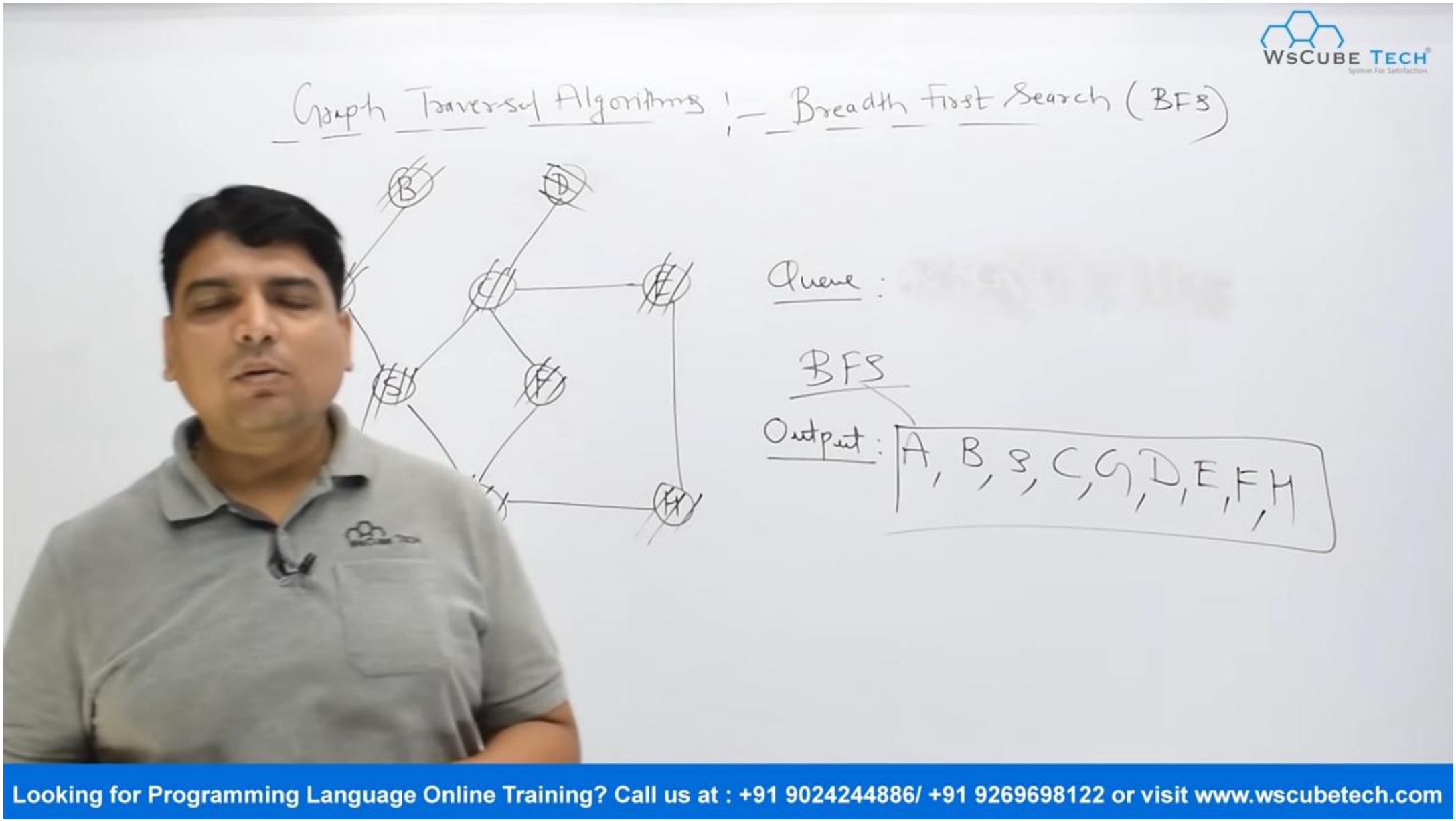
WsCube Tech
System For Satisfaction

Graph Traversal Algorithms :- Breadth First Search (BFS)



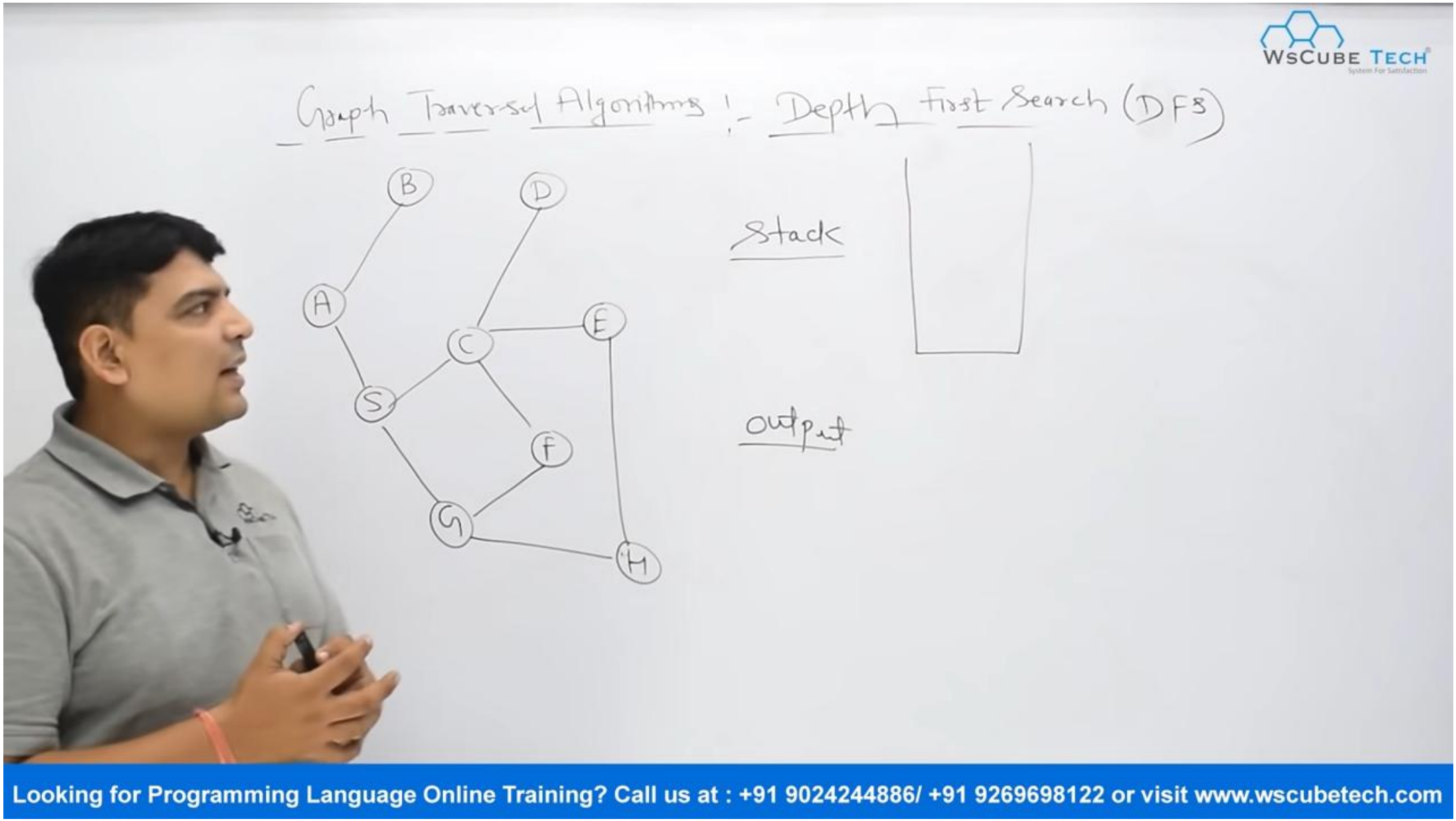
Queue :

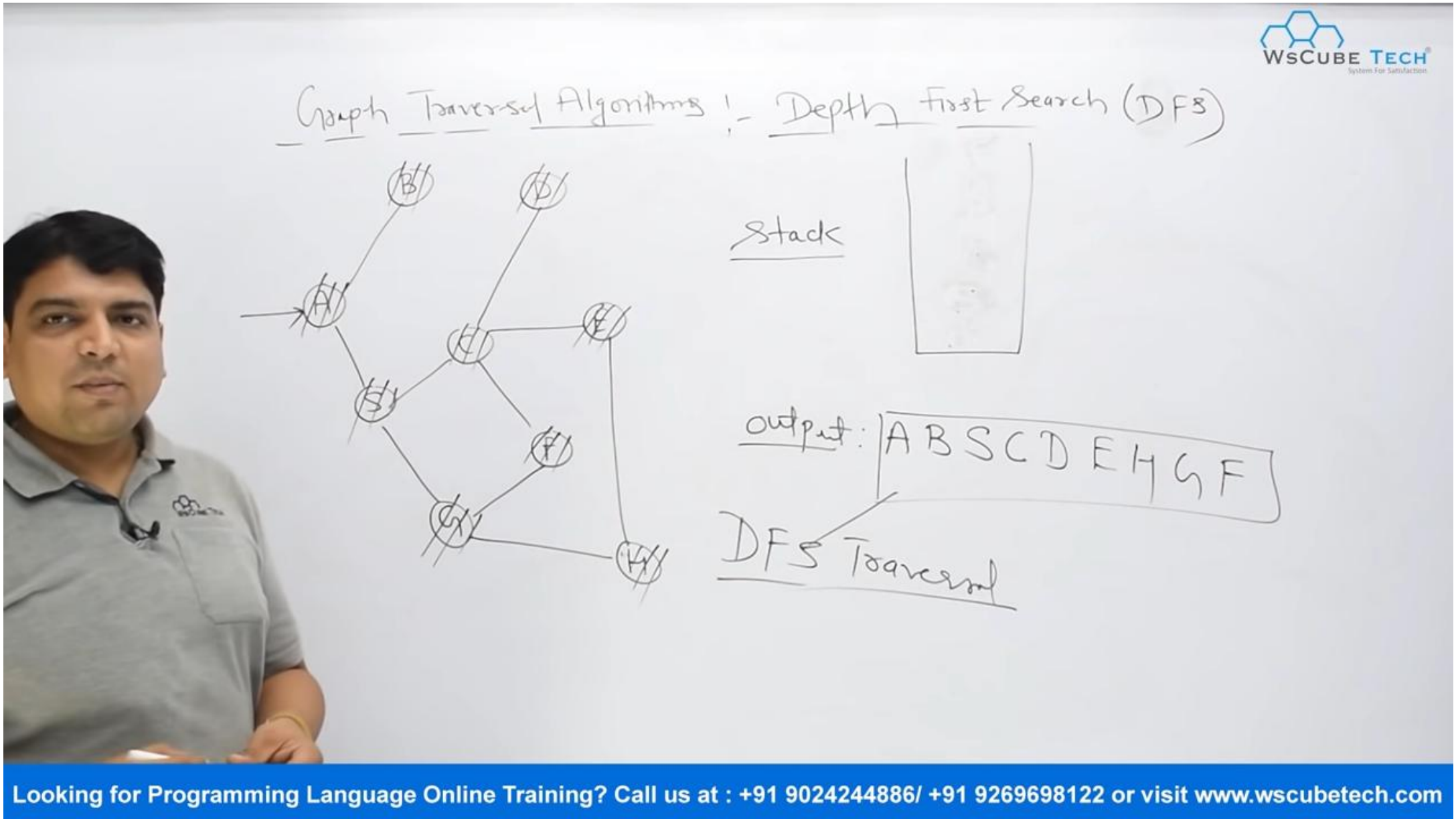
Output : A, B, S, C, G, D, E, F, H


WsCube Tech
System For Satisfaction
Graph Traversal Algorithms :- Breadth First Search (BFS)
Queue :
BFS
Output : A, B, S, C, G, D, E, F, H
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com

Data Structure & Algorithm

Graph Traversal - Depth Fisrt Search


WsCube Tech
System For Satisfaction
Graph Traversal Algorithms :- Depth First Search (DFS)
B
D
A
E
C
S
F
G
H
Stack
output
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com


WsCube Tech
System For Satisfaction
Graph Traversal Algorithms :- Depth First Search (DFS)
Stack
output : A B S C D E H G F
DFS Traversal
Looking for Programming Language Online Training? Call us at : +91 9024244886/ +91 9269698122 or visit www.wscubetech.com