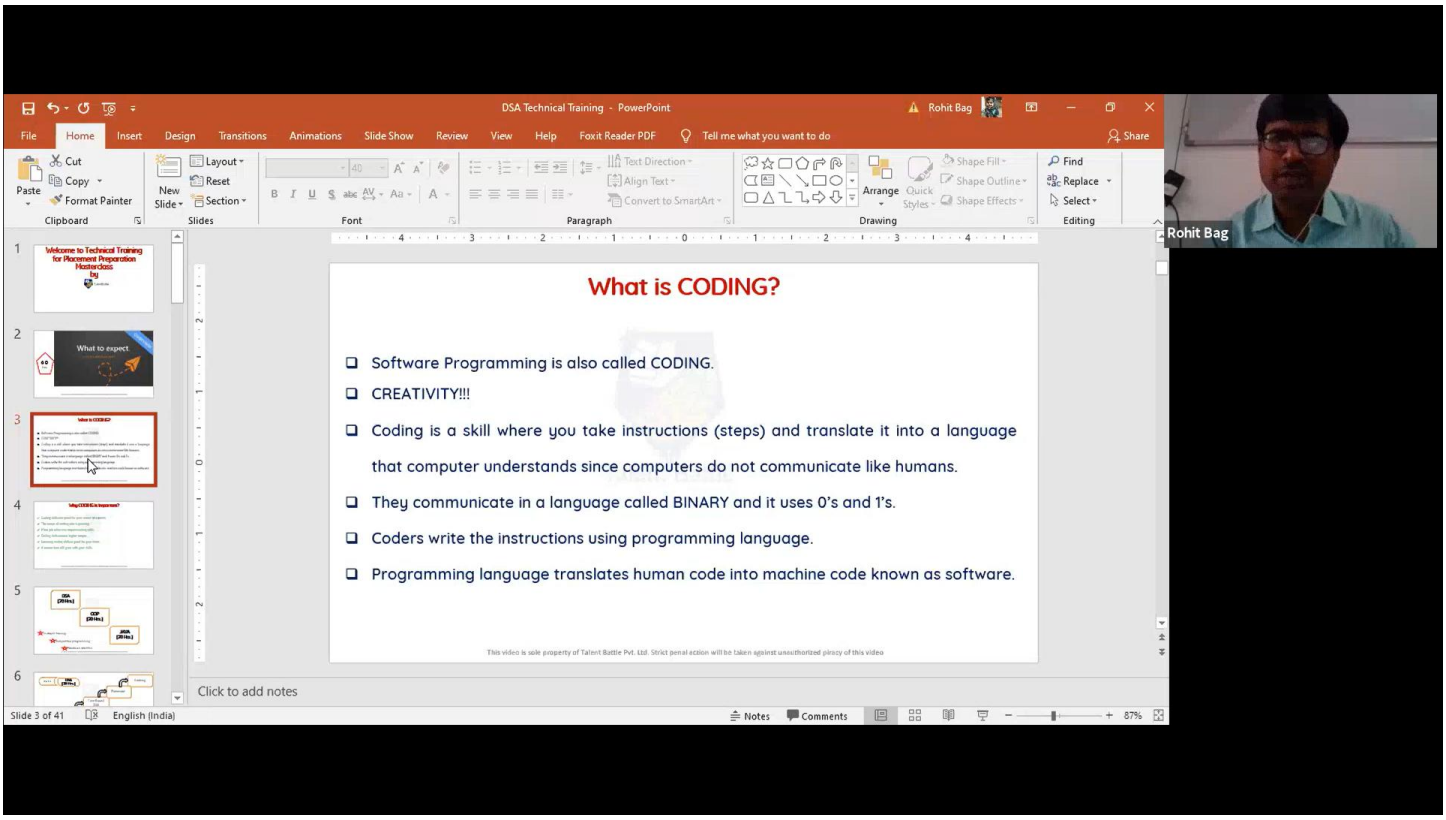
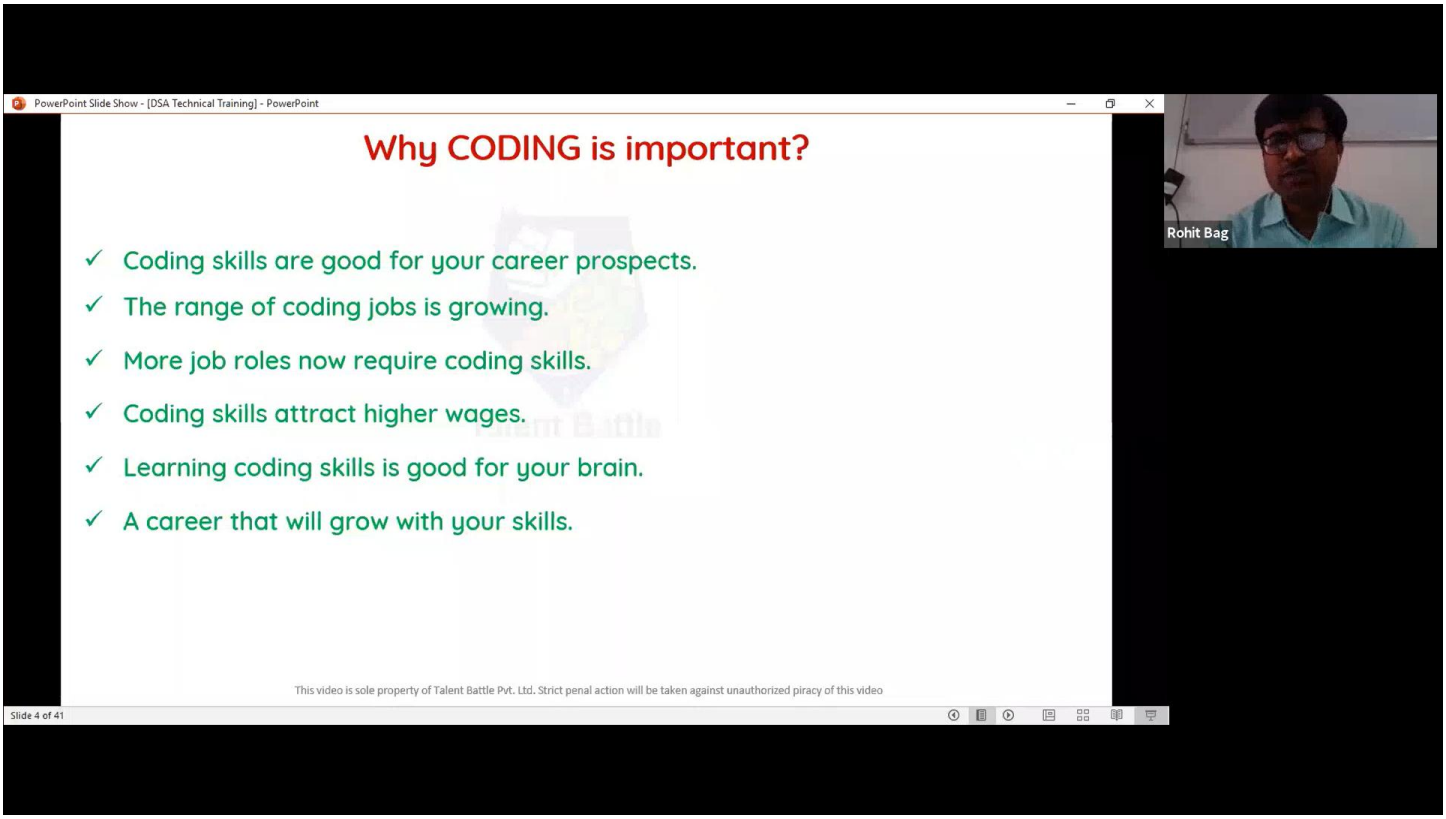
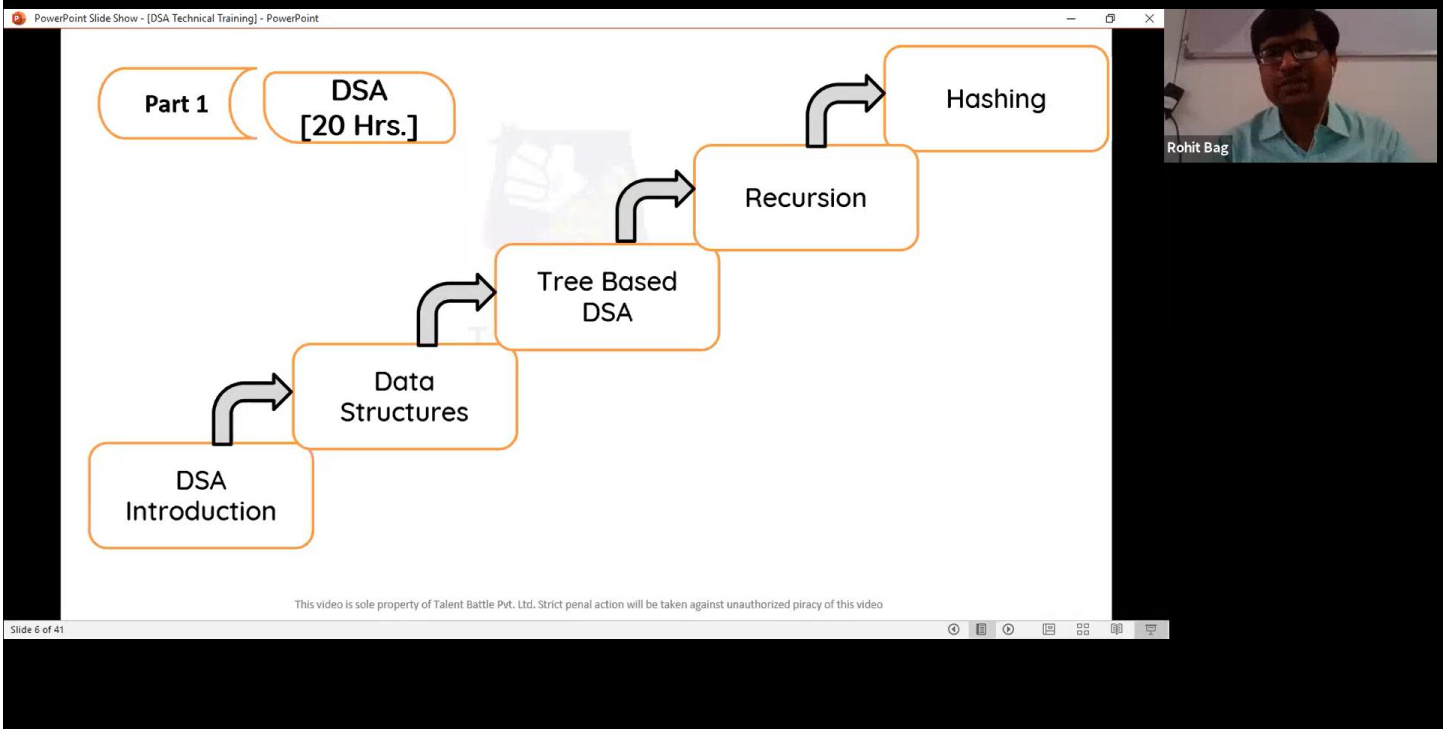
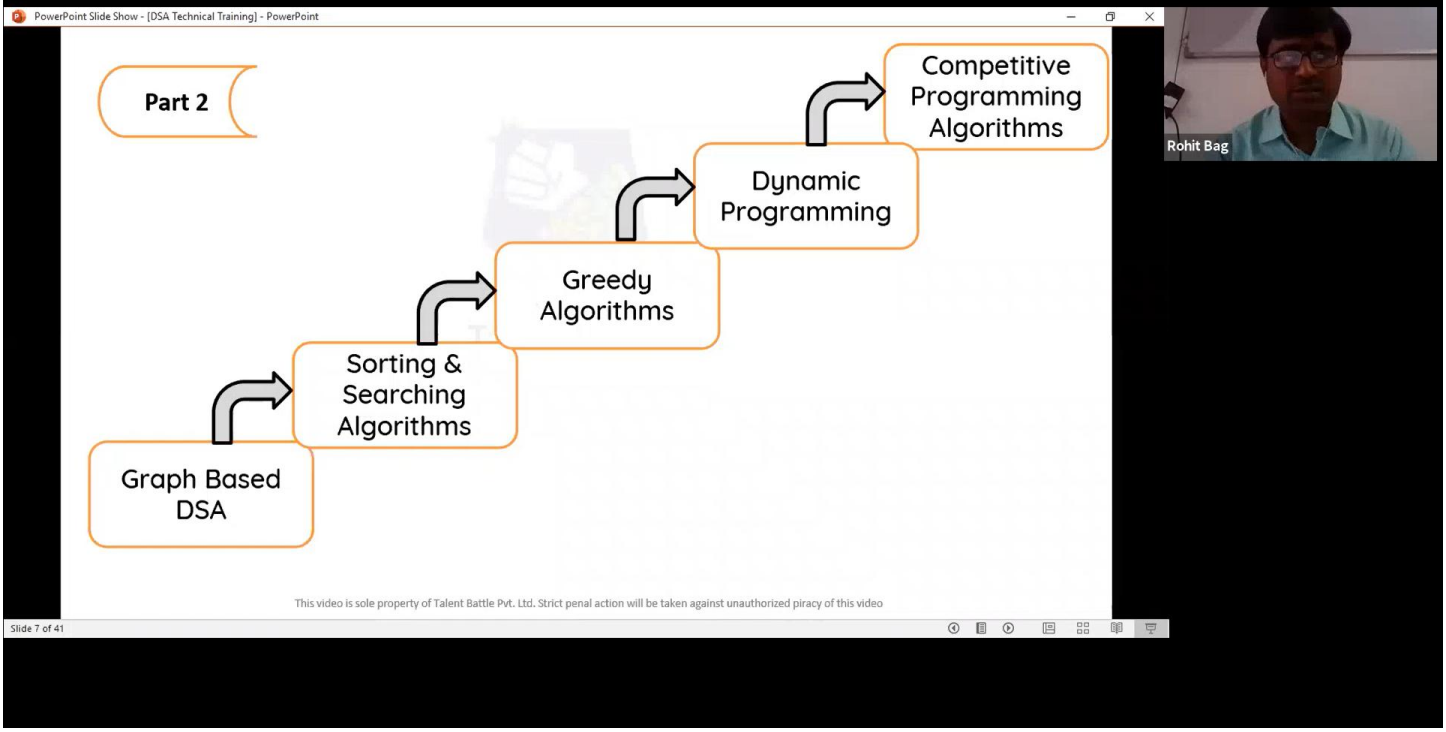
Day 1
Data Structures & Algorithms (using C++)
DSA Introduction Part 1




PowerPoint Slide Show - [DSA Technical Training] - PowerPoint
Part 1
DSA
[20 Hrs.]
DSA Introduction
Data Structures
Tree Based DSA
Recursion
Hashing
This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video
Slide 6 of 41
Rohit Bag


PowerPoint Slide Show - [DSA Technical Training] - PowerPoint
Part 2
Graph Based DSA
Sorting & Searching Algorithms
Greedy Algorithms
Dynamic Programming
Competitive Programming Algorithms
This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video
Slide 7 of 41
Rohit Bag

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Introduction

- A computer program is a collection of instructions to perform a specific task. For this, a computer program may need to store data, retrieve data, and perform computations on the data.
- A data structure is a named location that can be used to store and organize data.






PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Good Computer Program

- Run efficiently and correctly.
- User friendly interface.
- Easy to read and understand.
- Easy to debug.
- Easy to modify.
- Easy to maintain.





PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Good Computer Program

Program consists of 2 things:

a) Algorithms

b) Data Structures

*Good program is combination of both.

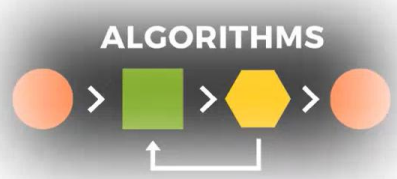


Slide 10 of 41


PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Algorithm??

- An **algorithm** is a precise procedure for solving a problem in finite number of steps.
- An **algorithm** states the actions to be executed and the order in which these actions are to be executed.





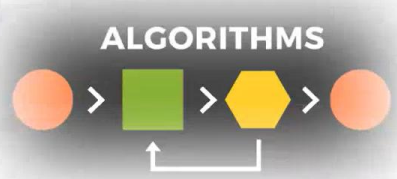

Slide 12 of 41


-1:35:29 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Algorithmic Properties

- It must be correct.
- Composed of series of concrete steps.
- No ambiguity.
- Composed of finite number of steps.
- Must terminate.
- It takes zero or more inputs.
- It should use less memory space as much as possible.
- It results in one or more outputs.






Slide 13 of 41


-1:34:07 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## How to develop algorithm?

- Defining the algorithm.
- Validating the algorithm.
- Expressing the algorithm.



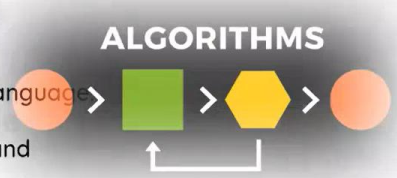

Slide 14 of 41


-1:28:40 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Efficiency of an algorithm.

- Algorithms are programs in general form.
- It is an idea upon which program is designed.
- It should be independent of the programming language.
- Every programmer having enough knowledge and experience should understand it.
- It should be applicable to inputs of all sizes.






Slide 15 of 41


-1:25:02 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Efficiency of an algorithm.

- Efficiency of an algorithm denotes the rate at which an algorithm solves a problem of size n.
- Time and Space
- Time = no. of steps algorithm executes.
- Space = no. of unit memory storage it requires.
- Algorithm complexity calculation: Time taken & Space required.




Slide 16 of 41


-1:22:07 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Characteristics of an algorithm.

- ❑ Unambiguous
- ❑ Input
- ❑ Output
- ❑ Finiteness
- ❑ Feasibility
- ❑ Independent



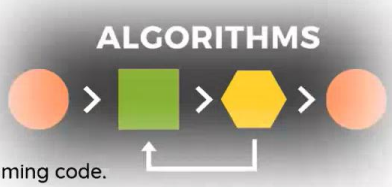




-1:18:39 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## How to write an Algorithm?




- ❑ It is problem and resource dependent.
- ❑ Algorithm are never written to support a particular programming code.
- ❑ Problem domain should be well defined.
- ❑ Example: Design an algorithm to add 2 numbers and display the result.

**Step 1** – START
**Step 2** – declare three integers **a**, **b** & **c**
**Step 3** – define values of **a** & **b**
**Step 4** – add values of **a** & **b**
**Step 5** – store output of step 4 to **c**
**Step 6** – print **c**
**Step 7** – STOP






-1:16:06 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Time Complexity of an algorithm.

- The time taken for an algorithm is comprised of:

  Compilation Time

  Run Time

- Compile time is the time taken to compile an algorithm.

  It checks syntax and semantic errors.

- Run time is the time to execute the compiled program.

Is runtime calculated only for executable statements or declaration statements?







-1:15:36 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Time Complexity of an algorithm.

- Time complexity of an algorithm is generally classified as three types:

  Worst Case

  Average Case

  Best Case







-57:41 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Time Complexity of an algorithm.

- **Worst Case**: Longest time that an algorithm will use to produce desired results.
- **Average Case** : Average time that the algorithm will use. It depends upon probability distribution.
- **Best Case**: Shortest time that the algorithm will use to produce desired results.







-51:50 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Space Complexity of an algorithm.

- Space complexity of the program is the amount of memory consumed.
- Fixed amount of memory.
- Variable amount of memory.
- Memory taken by instruction not in control of programmer.
- Memory taken by variable is in control of programmer.

  Datatype Array[size];







-51:23 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Space Complexity of an algorithm.

- There are 3 different spaces considered for determining the amount of memory used by the algo.

1. **Instruction Space**: occupied by compile version of program.
2. **Data Space**: used to hold variables, data structures, data elements.
3. **Environment Space**: used for run time stack on function call.




Slide 23 of 41


-47:53 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Problem Solving??

- A Programmer should identify all the requirements to solve the problem:

✓ Type of programming language
✓ Narration of the program describing tasks to be performed
✓ Frequency of processing
✓ Input & Output of the program
✓ Limitations & restrictions for the program
✓ Detailed specifications

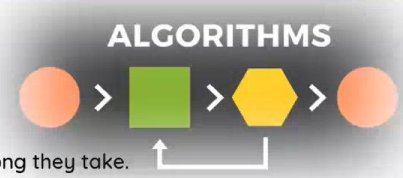



Slide 24 of 41


-42:27 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Algorithm Analysis





- **Method 1**: Code each algorithm and run them to see how long they take.
  - **Problem**: How will you know there is better program?
- **Method 2**: Develop a model of the way computers work and compare how the algorithm behaves in the model.
- Able to distinguish between good and bad algorithms.



Slide 25 of 41

-30:15 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## How to measure algorithm performance?


- Length of the program (LOC)
- Ease of programming (bugs, maintenance)
- Memory required
- Running time




Slide 26 of 41

-26:40 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Importance of studying Algorithm?

- Interview Preparations : provides competitive advantage during job search process.
- Keeping your programming skills sharp.

"Practice makes Perfect"


- Long term career goals.

**"Practicing algorithms will increase your skill and your visibility at work".**



Slide 27 of 41


-21:32 1x

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

## Summary

- Introduction
- Good Computer Program
- DSA Introduction
- Algorithm
- Algo. Properties
- How to develop algo.
- Efficiency of algo.
- Characteristics of algo.
- How to write an algo.
- Time complexity
- Space complexity
- Problem solving
- Algorithm Analysis
- How to measure algo. Performance
- Importance of studying algorithm




Slide 28 of 41