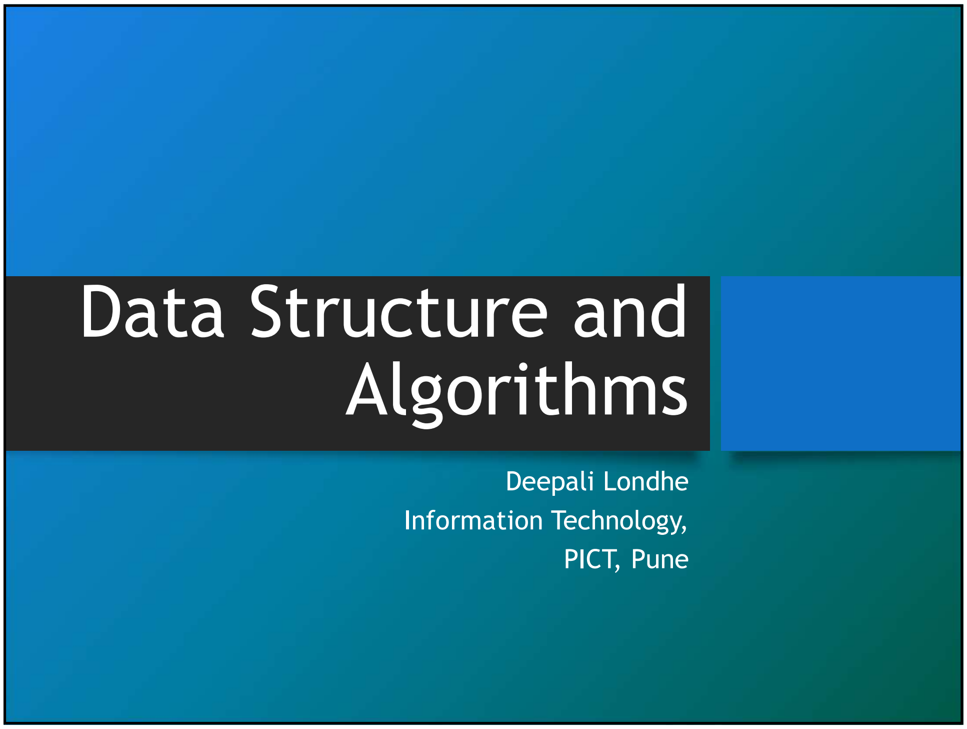
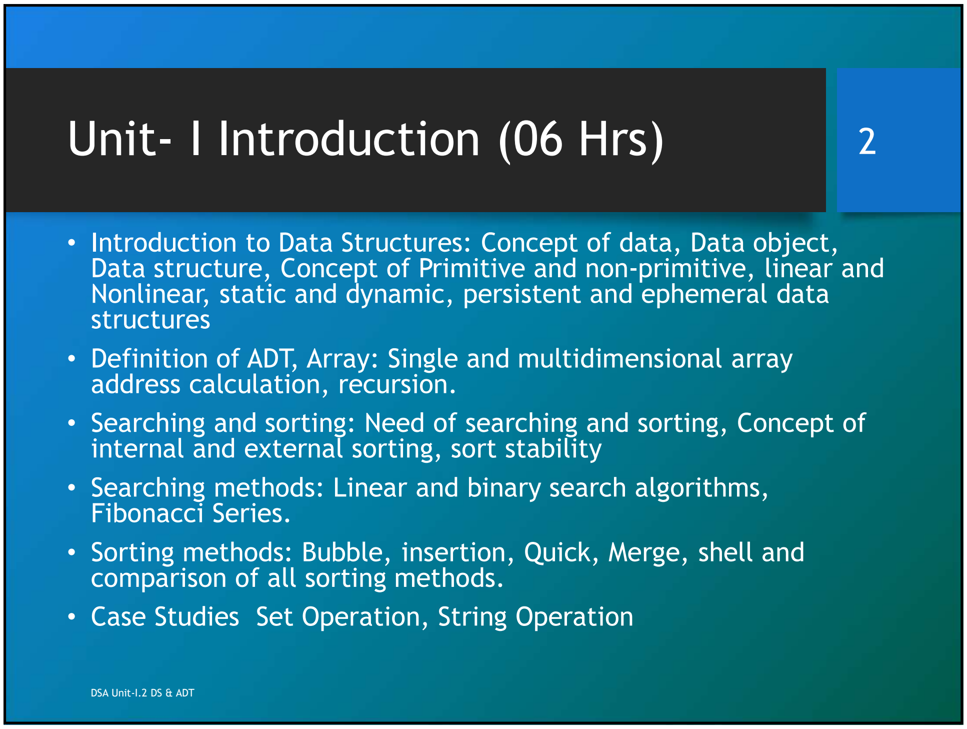


Data Structure and Algorithms

Deepali Londhe
Information Technology,
PICT, Pune

1



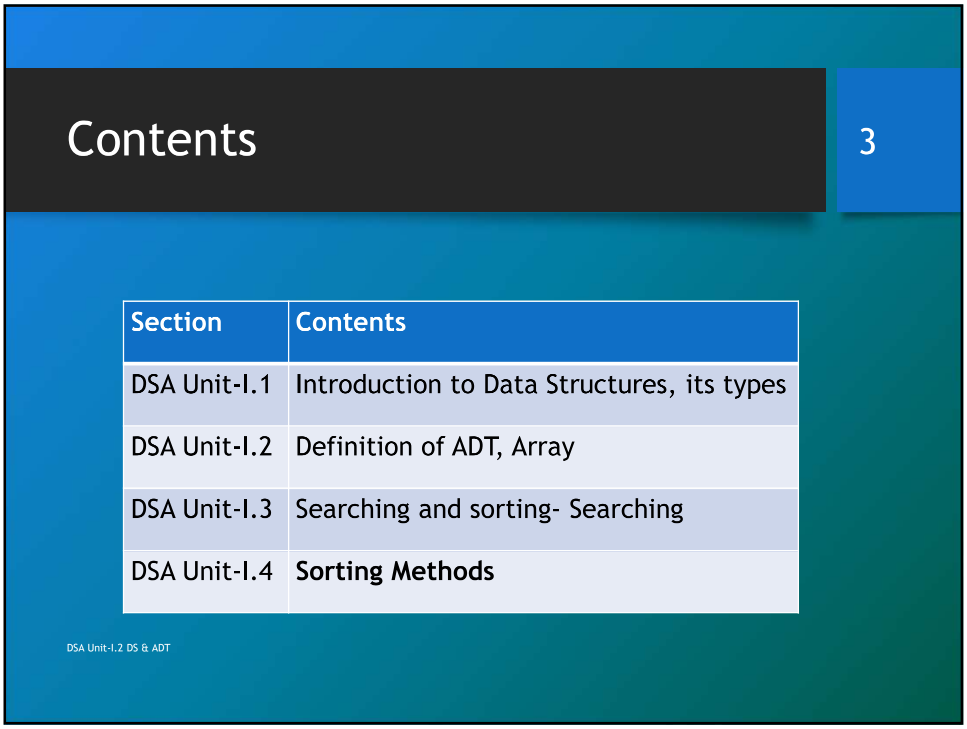
Unit- I Introduction (06 Hrs)

2

- Introduction to Data Structures: Concept of data, Data object, Data structure, Concept of Primitive and non-primitive, linear and Nonlinear, static and dynamic, persistent and ephemeral data structures
- Definition of ADT, Array: Single and multidimensional array address calculation, recursion.
- Searching and sorting: Need of searching and sorting, Concept of internal and external sorting, sort stability
- Searching methods: Linear and binary search algorithms, Fibonacci Series.
- Sorting methods: Bubble, insertion, Quick, Merge, shell and comparison of all sorting methods.
- Case Studies Set Operation, String Operation

DSA Unit-I.2 DS & ADT

2

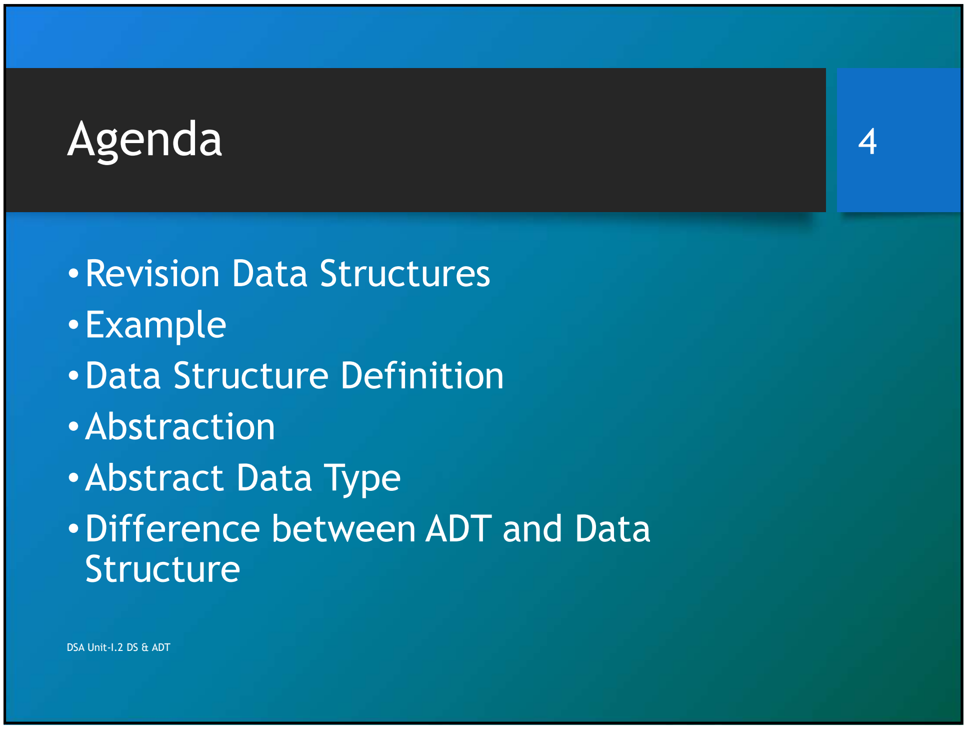


Contents

3

Section	Contents
DSA Unit-I.1	Introduction to Data Structures, its types
DSA Unit-I.2	Definition of ADT, Array
DSA Unit-I.3	Searching and sorting- Searching
DSA Unit-I.4	**Sorting Methods**

3

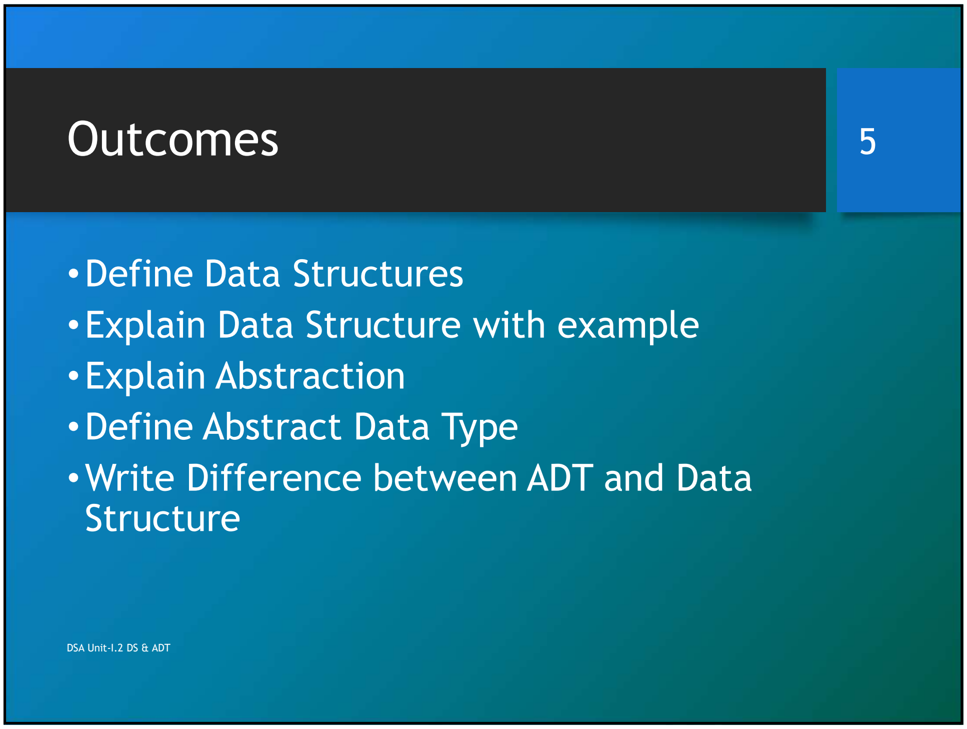


Agenda

4

- Revision Data Structures
- Example
- Data Structure Definition
- Abstraction
- Abstract Data Type
- Difference between ADT and Data Structure

4



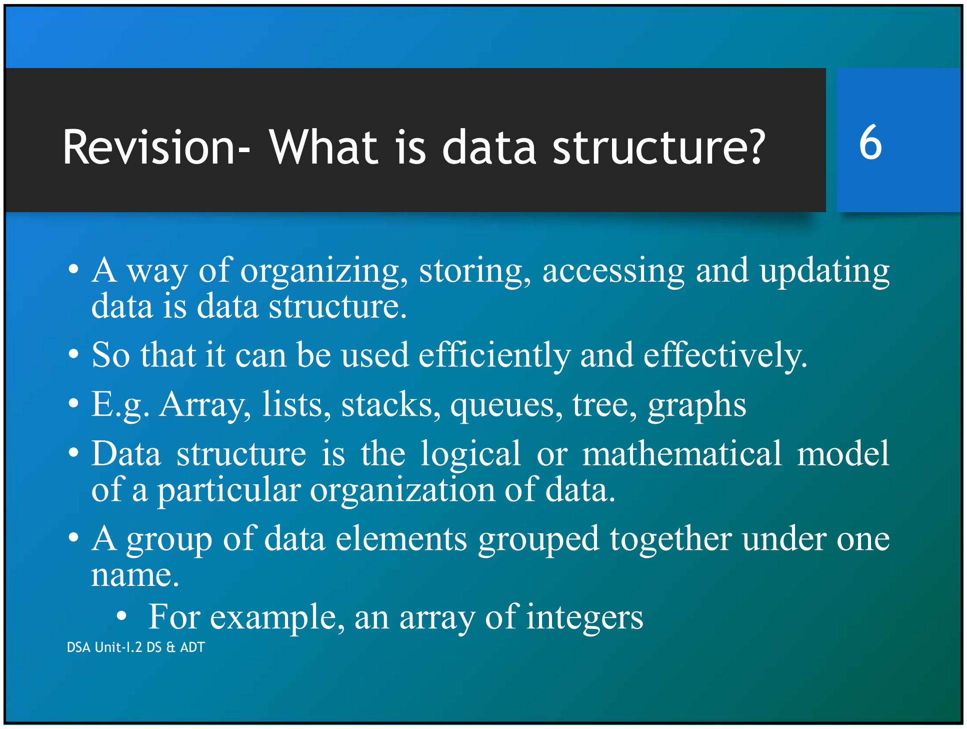
Outcomes

5

- Define Data Structures
- Explain Data Structure with example
- Explain Abstraction
- Define Abstract Data Type
- Write Difference between ADT and Data Structure

DSA Unit-I.2 DS & ADT

5



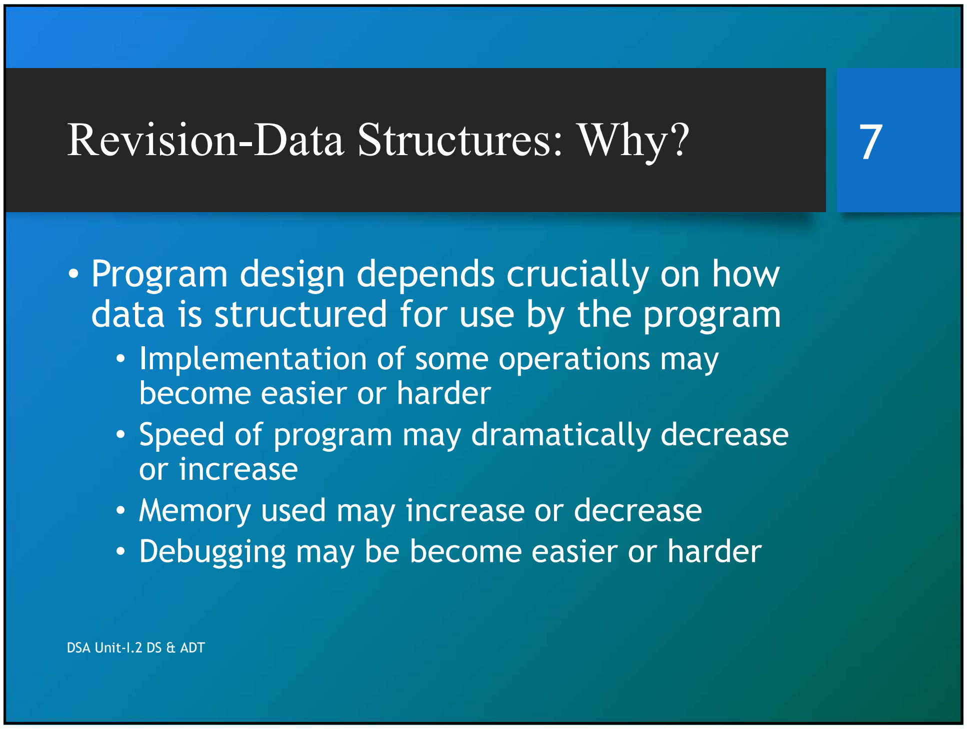
Revision- What is data structure?

6

- A way of organizing, storing, accessing and updating data is data structure.
- So that it can be used efficiently and effectively.
- E.g. Array, lists, stacks, queues, tree, graphs
- Data structure is the logical or mathematical model of a particular organization of data.
- A group of data elements grouped together under one name.
  - For example, an array of integers

DSA Unit-I.2 DS & ADT

6

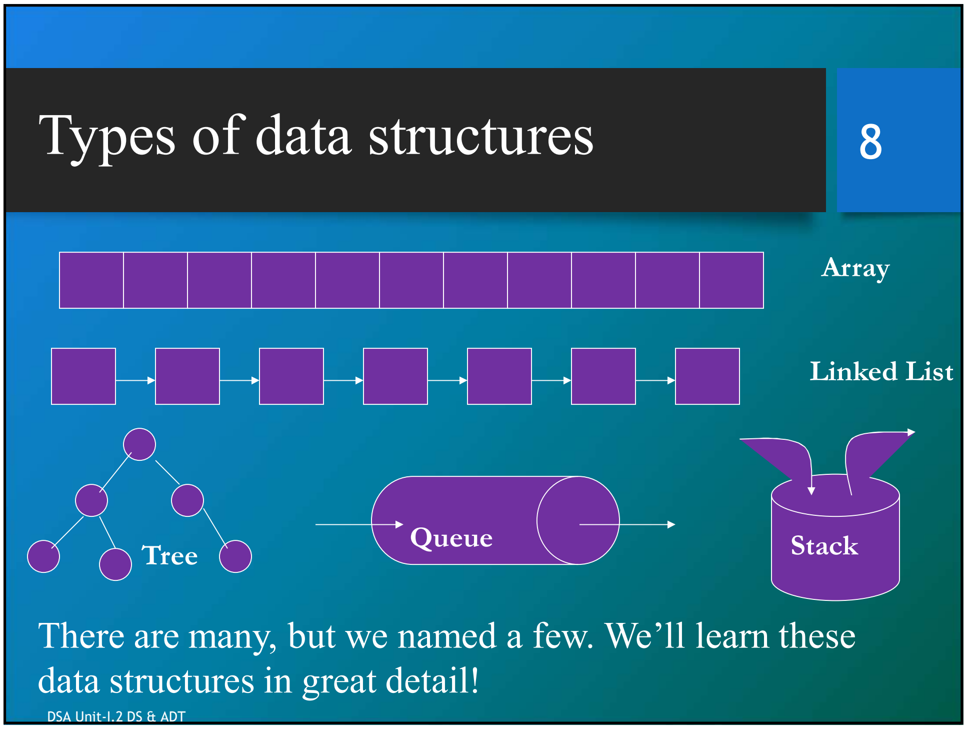


Revision-Data Structures: Why?

7

- Program design depends crucially on how data is structured for use by the program
  - Implementation of some operations may become easier or harder
  - Speed of program may dramatically decrease or increase
  - Memory used may increase or decrease
  - Debugging may be become easier or harder

7



Types of data structures

8

Array

Linked List

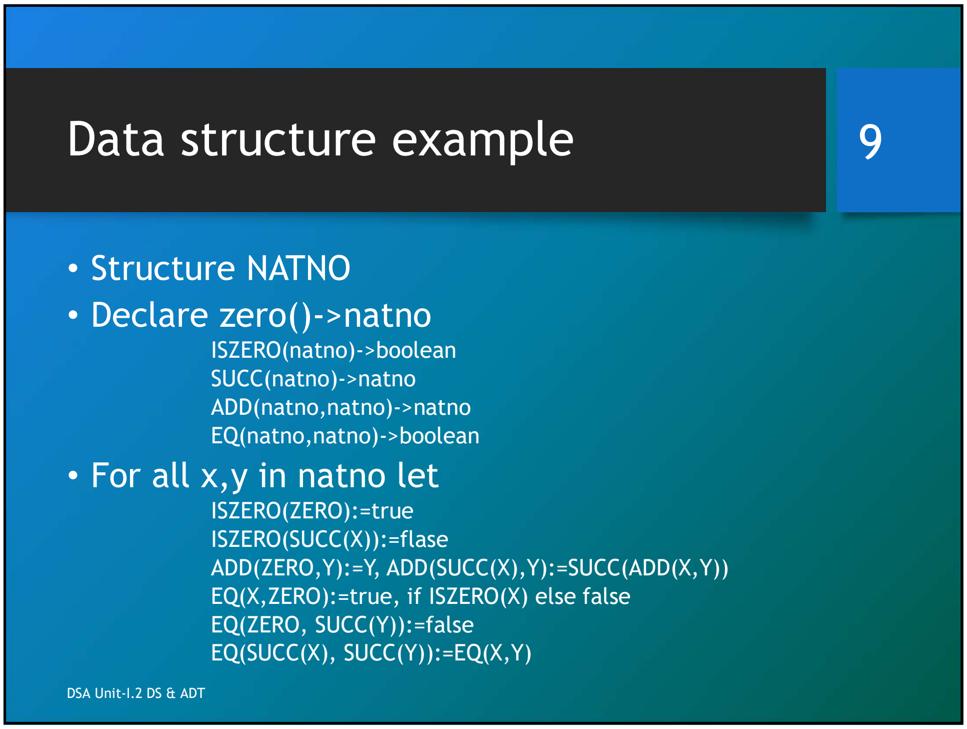
Tree

Queue

Stack

There are many, but we named a few. We'll learn these data structures in great detail!

8



Data structure example

9

- Structure NATNO
- Declare zero()->natno

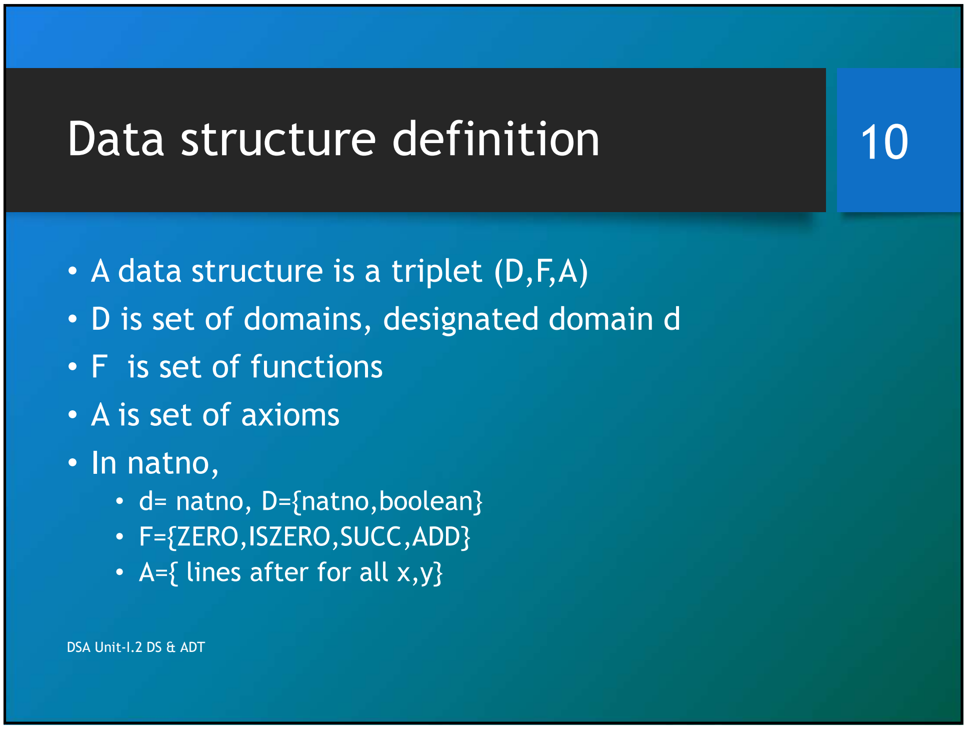
ISZERO(natno)->boolean
SUCC(natno)->natno
ADD(natno,natno)->natno
EQ(natno,natno)->boolean

- For all x,y in natno let

ISZERO(ZERO):=true
ISZERO(SUCC(X)):=flase
ADD(ZERO,Y):=Y, ADD(SUCC(X),Y):=SUCC(ADD(X,Y))
EQ(X,ZERO):=true, if ISZERO(X) else false
EQ(ZERO, SUCC(Y)):=false
EQ(SUCC(X), SUCC(Y)):=EQ(X,Y)

DSA Unit-I.2 DS & ADT

9



Data structure definition

10

- A data structure is a triplet (D,F,A)
- D is set of domains, designated domain d
- F is set of functions
- A is set of axioms
- In natno,
  - d= natno, D={natno,boolean}
  - F={ZERO,ISZERO,SUCC,ADD}
  - A={ lines after for all x,y}

DSA Unit-I.2 DS & ADT

10

Abstraction

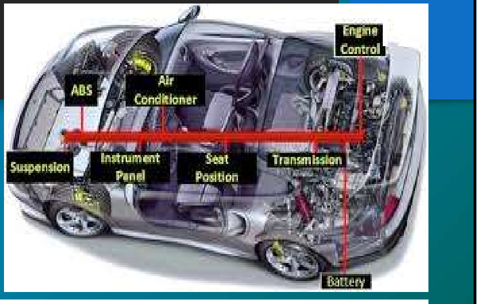
11

- Hiding unnecessary details is known as **abstraction.**
- Only presenting an interface, not the implementation part .
- i.e. only an interface is shown and implementation part is hidden .
- An essential element of object oriented programming is **abstraction.**

DSA Unit-I.2 DS & ADT

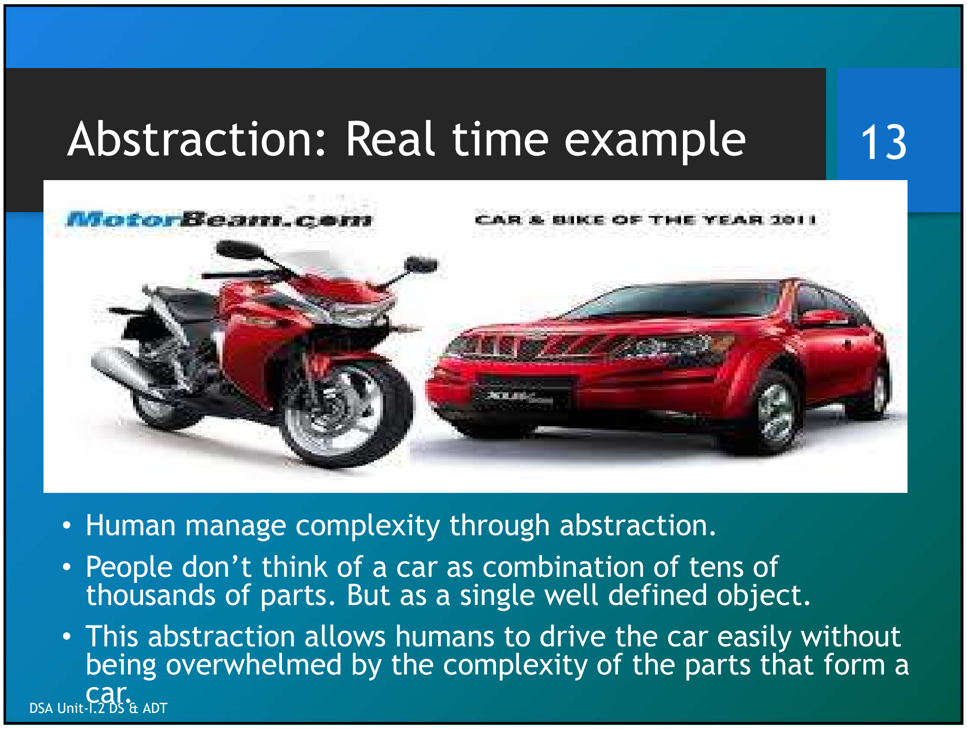
11

Abstraction: Real time example



Unit-I.2 DS & ADT

12



MotorBeam.com
CAR & BIKE OF THE YEAR 2011

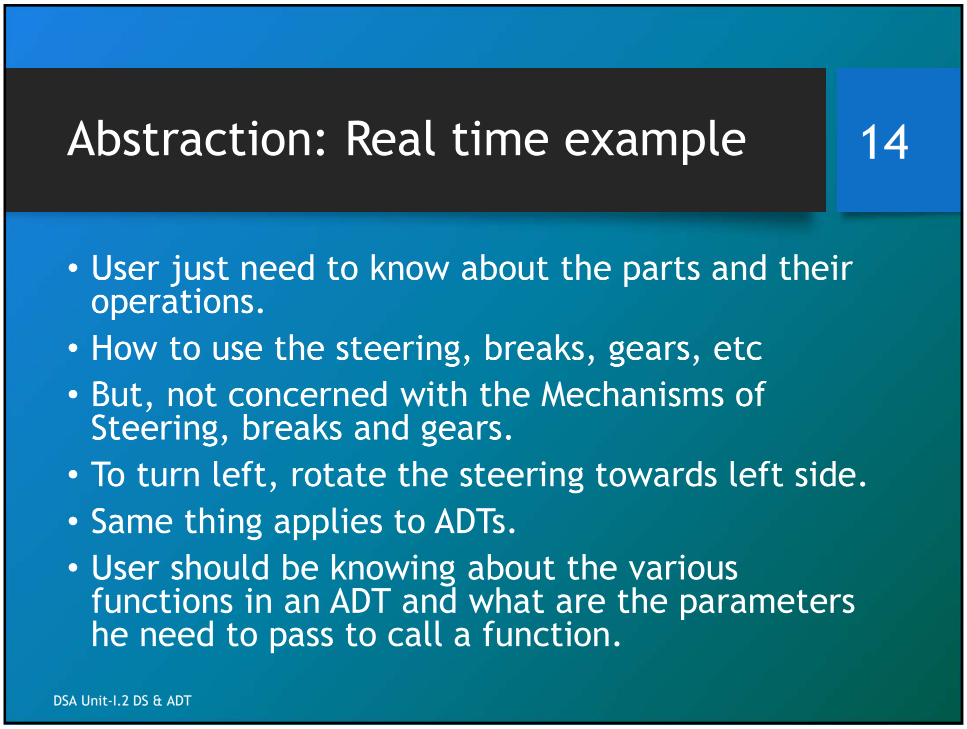
Abstraction: Real time example

13

- Human manage complexity through abstraction.
- People don’t think of a car as combination of tens of thousands of parts. But as a single well defined object.
- This abstraction allows humans to drive the car easily without being overwhelmed by the complexity of the parts that form a car.

DSA Unit-I.2 DS & ADT

13



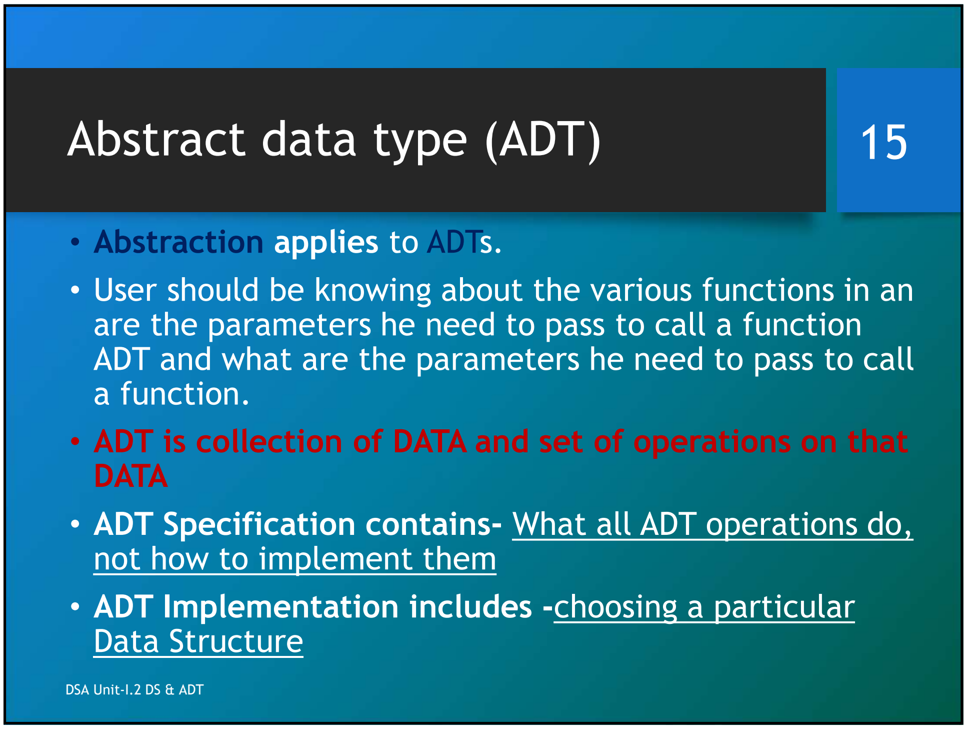
Abstraction: Real time example

14

- User just need to know about the parts and their operations.
- How to use the steering, breaks, gears, etc
- But, not concerned with the Mechanisms of Steering, breaks and gears.
- To turn left, rotate the steering towards left side.
- Same thing applies to ADTs.
- User should be knowing about the various functions in an ADT and what are the parameters he need to pass to call a function.

DSA Unit-I.2 DS & ADT

14



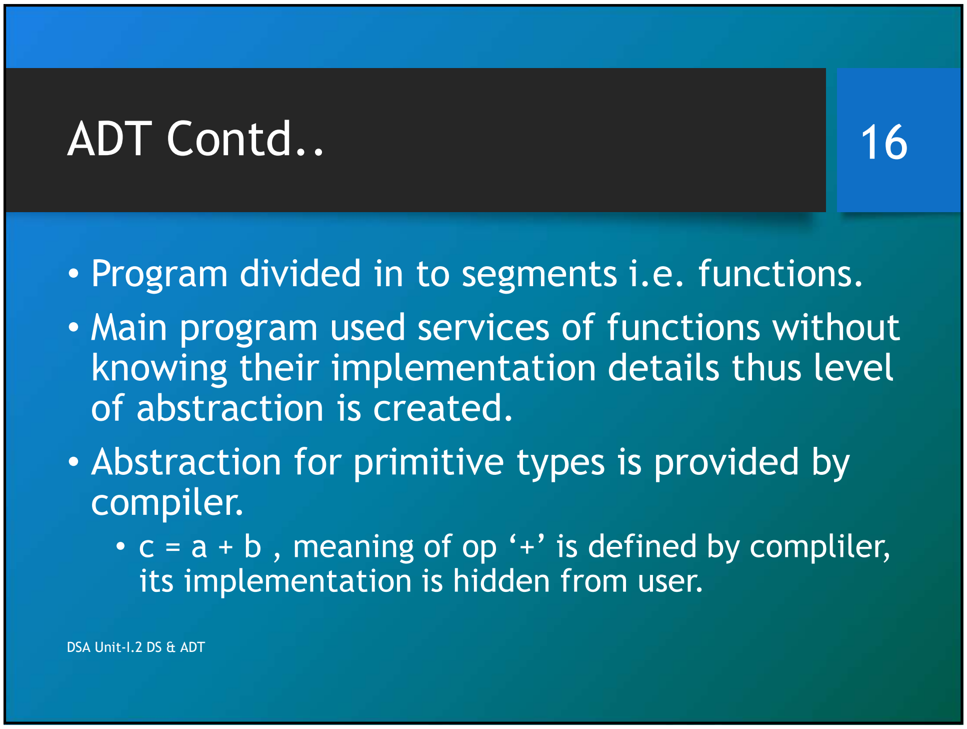
Abstract data type (ADT)

15

- **Abstraction applies** to ADTs.
- User should be knowing about the various functions in an are the parameters he need to pass to call a function ADT and what are the parameters he need to pass to call a function.
- **ADT is collection of DATA and set of operations on that DATA**
- **ADT Specification contains-** What all ADT operations do, not how to implement them
- **ADT Implementation includes -**choosing a particular Data Structure

DSA Unit-I.2 DS & ADT

15



ADT Contd..

16

- Program divided in to segments i.e. functions.
- Main program used services of functions without knowing their implementation details thus level of abstraction is created.
- Abstraction for primitive types is provided by compiler.
  - c = a + b , meaning of op ‘+’ is defined by compliler, its implementation is hidden from user.

DSA Unit-I.2 DS & ADT

16

ADT for set operations

17

- a,b,c are sets
- n1,n2,n3 are no of elements in set a, b, c respectivly
- Inputset(a,b,n1,n2): get user input
- unionset(a,b,c,n1,n2): this returns size of set c in n3
- intersection(a,b,c,n1,n2): returns size after intersection in n3
- difference(a,b,c,n1,n2): return n3
- symdifference(a,b,c,n1,n2): return n3
- display(a,n1): display set a of size n1

ADT=Data Structure?

18

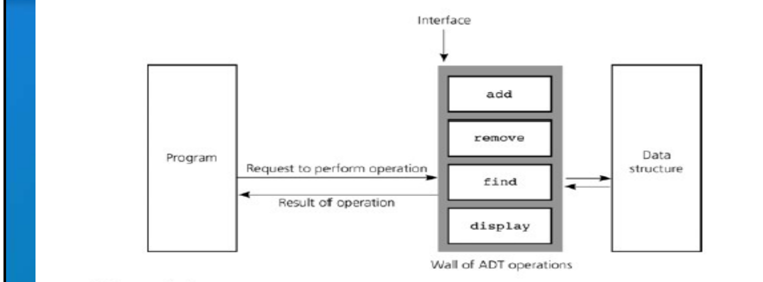
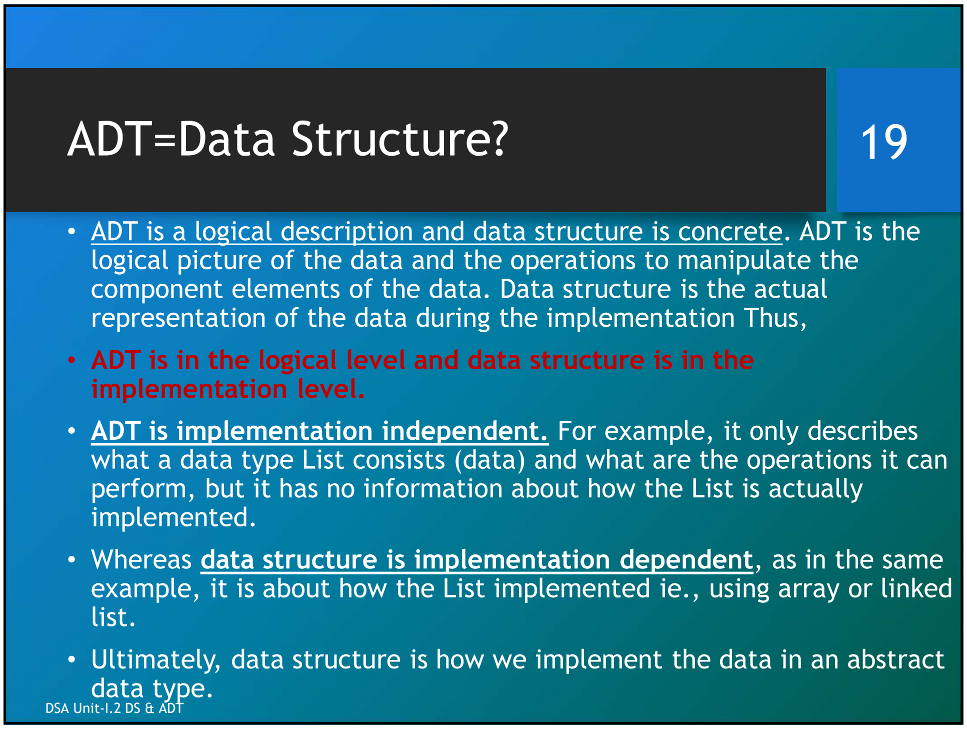


Figure 4-4
A wall of ADT operations isolates a data structure from the program that uses it

Data Structure- A construct that is defined in a program language to store collection of data. Examples: arrays

ADTs and Data Structures are not the same.
Data Abstraction: Results in wall of ADT operations between data structures and the program that access the data within this data structure.



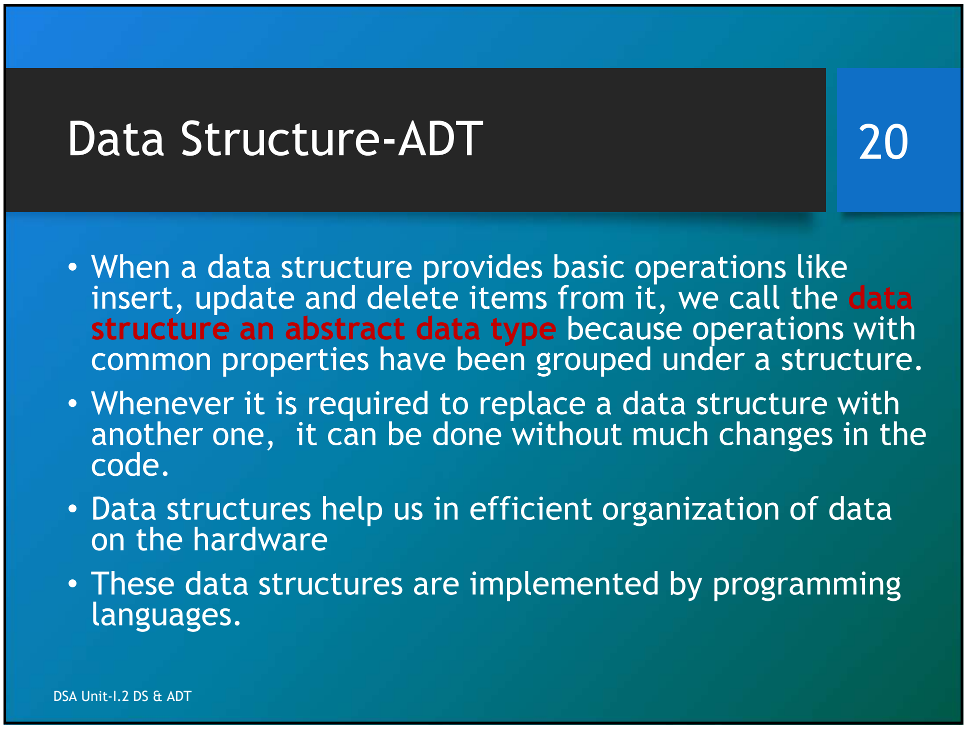
ADT=Data Structure?

19

- ADT is a logical description and data structure is concrete. ADT is the logical picture of the data and the operations to manipulate the component elements of the data. Data structure is the actual representation of the data during the implementation Thus,
- **ADT is in the logical level and data structure is in the implementation level.**
- **ADT is implementation independent.** For example, it only describes what a data type List consists (data) and what are the operations it can perform, but it has no information about how the List is actually implemented.
- Whereas **data structure is implementation dependent**, as in the same example, it is about how the List implemented ie., using array or linked list.
- Ultimately, data structure is how we implement the data in an abstract data type.

DSA Unit-I.2 DS & ADT

19



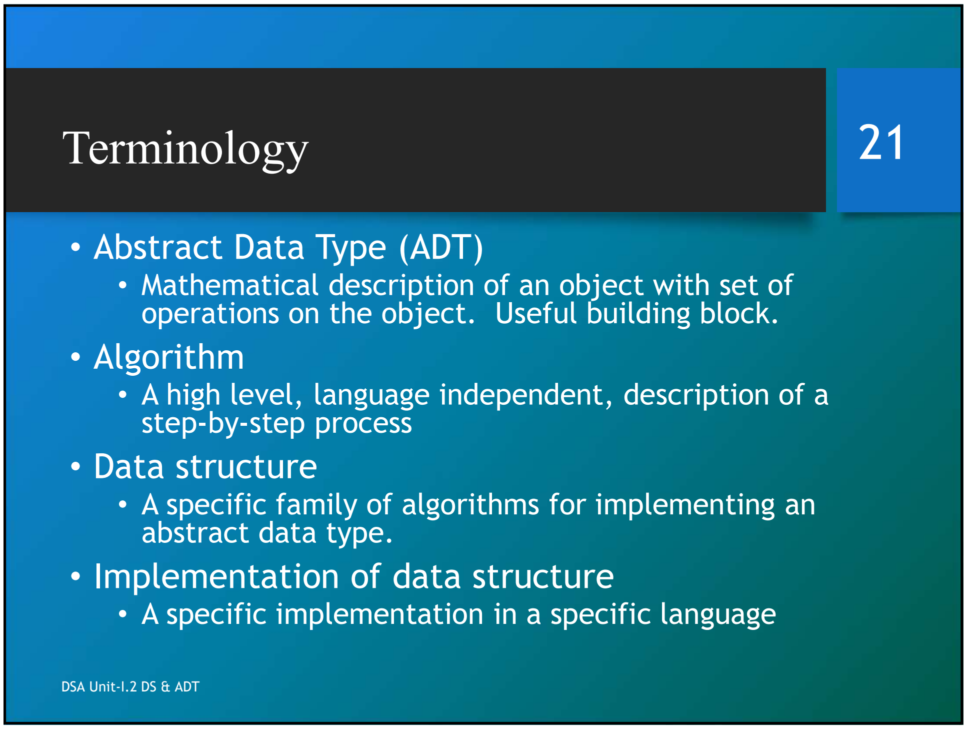
Data Structure-ADT

20

- When a data structure provides basic operations like insert, update and delete items from it, we call the data structure an abstract data type because operations with common properties have been grouped under a structure.
- Whenever it is required to replace a data structure with another one, it can be done without much changes in the code.
- Data structures help us in efficient organization of data on the hardware
- These data structures are implemented by programming languages.

DSA Unit-I.2 DS & ADT

20



Terminology

21

- Abstract Data Type (ADT)
  - Mathematical description of an object with set of operations on the object. Useful building block.
- Algorithm
  - A high level, language independent, description of a step-by-step process
- Data structure
  - A specific family of algorithms for implementing an abstract data type.
- Implementation of data structure
  - A specific implementation in a specific language

DSA Unit-I.2 DS & ADT

21

References

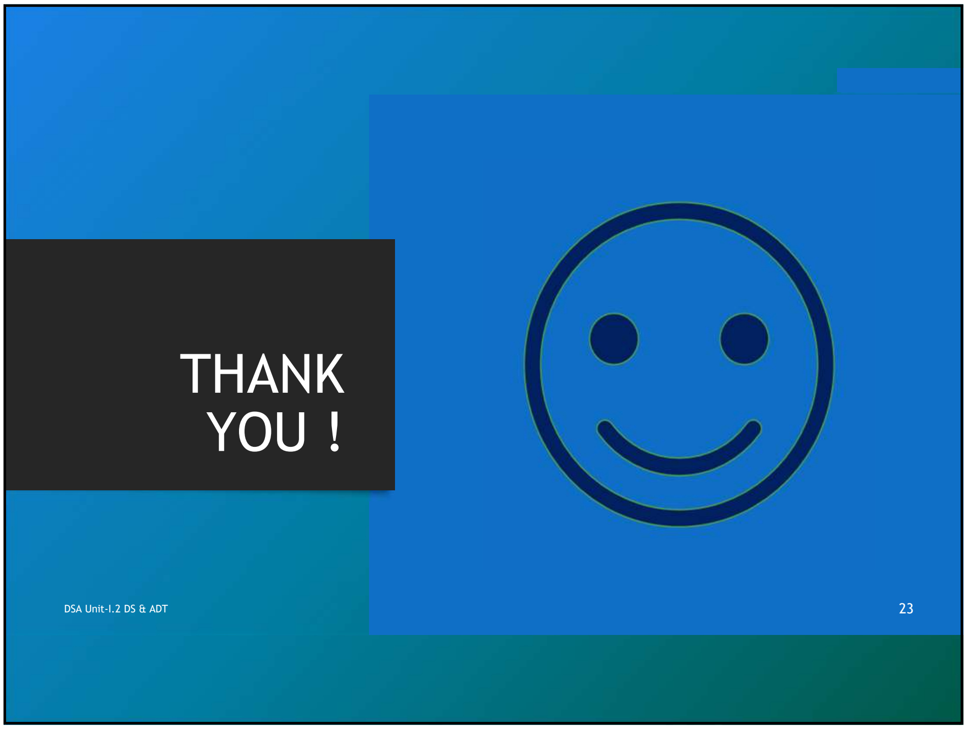
22

- **Books**
- Ellis Horowitz and Sartaj Sahni , “Fundamentals of Data Structures”, Computer Science Press, 1983
- R. Gilberg, B. Forouzan, "Data Structures: A pseudo Code Approach with C++", Cengage Learning, ISBN 9788131503140.
- E. Horowitz, S. Sahni, D. Mehta, "Fundamentals of Data Structures in C++", Galgotia Book Source, New Delhi, 1995, ISBN 16782928
- Dinesh P. Shah, Sartaj Sahani , “Handbook of DATA STRUCTURES and APPLICATIONS”, CHAPMAN & HALL/CRC
- Bayer B. et al. (2015) Electro-Mechanical Brake Systems. In: Winner H., Hakuli S., Lotz F., Singer C. (eds) Handbook of Driver Assistance Systems. Springer, Cham
- **Web**
  - http://statmath.wu.ac.at/courses/data-analysis/itdtHTML/node55.html
  - https://en.wikipedia.org/wiki/Persistent_data_structure

No copyright infringement is intended

DSA Unit-I.2 DS & ADT

22



THANK YOU !

23