

# TOPPER NOTES

**SUBJECT:**  
**DATA STRUCTURE USING C**

**COURSE:**  
**DIPLOMA IN INFORMATION TECHNOLOGY**

**YEAR / SEMESTER:**  
**SECOND YEAR – FOURTH SEMESTER**

**BOARD:**  
**BOARD OF TECHNICAL EDUCATION,  
UTTAR PRADESH**

**INSTITUTE:**  
**GOVERNMENT POLYTECHNIC  
AURAI, BHADOHI**

**PREPARED BY:**  
**NAME: AMAN JAISWAL**  
**ENROLLMENT NO: XXXXXXXX23**  
**BRANCH: INFORMATION TECHNOLOGY**

**SESSION:**  
**2025 – 2026**

**(Prepared for Academic Examination Purpose)**

## INDEX

S.No.	Unit No.	Title / Topics	Page No.
1	UNIT - 1	Introduction to Data Structures <ul style="list-style-type: none"><li>• Basic Terminology</li><li>• Classification of Data Structures</li><li>• Operations on Data Structures</li></ul>	_____
2	UNIT - 2	Linear Data Structures <ul style="list-style-type: none"><li>(A) Stacks<ul style="list-style-type: none"><li>• Introduction to Stacks</li><li>• Array Representation of Stacks</li><li>• Operations on Stack</li><li>• Applications of Stacks<ul style="list-style-type: none"><li>- Infix to Postfix Transformation</li><li>- Evaluation of Postfix Expression</li></ul></li></ul></li><li>(B) Queues<ul style="list-style-type: none"><li>• Introduction to Queues</li><li>• Array Representation of Queues</li><li>• Operations on Queue</li><li>• Types of Queues<ul style="list-style-type: none"><li>- DeQueue</li><li>- Circular Queue</li></ul></li><li>• Applications of Queues</li></ul></li></ul>	_____
3	UNIT - 3	Linked Lists <ul style="list-style-type: none"><li>• Singly Linked List</li><li>• Representation of Linked List in Memory</li><li>• Operations on Singly Linked List</li><li>• Circular Linked List</li><li>• Doubly Linked List</li></ul>	_____
4	UNIT - 4	Sorting and Searching <ul style="list-style-type: none"><li>(A) Sorting Techniques<ul style="list-style-type: none"><li>• Bubble Sort</li></ul></li></ul>	_____

- Insertion Sort
- Quick Sort
- Merge Sort

#### (B) Searching Techniques

- Symbol Tables
- Linear Search
- Binary Search
- Hash Tables

### 5 UNIT - 5 Trees and Graphs

---

#### (A) Trees

- Basic Terminologies
- Definition and Concepts of Binary Trees
- Representation of Binary Tree
  - Array Representation
  - Linked List Representation
- Operations on Binary Tree
  - Insertion
  - Deletion
  - Traversals
- Types of Binary Trees
- Binary Tree Traversals
  - In-order
  - Pre-order
  - Post-order
- Binary Search Tree
- B-Tree
- AVL Tree

#### (B) Graph

- Graph Terminologies
- Representation of Graphs
  - Adjacency Matrix
  - Adjacency List
- Graph Traversals

# INTRODUCTION TO DATA STRUCTURES

(Unit-1 : Introduction to Data Structures)

## 1 Definition (परिभाषा – point-wise)

1. **Data Structure** वह तकनीक (Technique) है जिसके द्वारा data को computer memory में **व्यवस्थित (Organized)** और **संरचित (Structured)** रूप में संग्रहित (Store) किया जाता है।
2. Data Structure का उपयोग data को **efficient** तरीके से **access**, **process** और **modify** करने के लिए किया जाता है।
3. यह data और algorithm के बीच **समन्वय (Coordination)** स्थापित करता है।
4. Proper Data Structure program की **efficiency (दक्षता)** और **performance (प्रदर्शन)** को बढ़ाता है।

## 2 Concept / Explanation (विस्तार से – 10–12 बिंदु)

1. Computer system में data को raw form में रखना उपयोगी नहीं होता, इसलिए उसे organized form में रखा जाता है।
2. Data Structure data elements के बीच **logical relationship (तार्किक संबंध – logical relation)** को दर्शाता है।
3. यह यह निर्धारित करता है कि data memory में कैसे store होगा और कैसे access होगा।
4. Different problems के लिए अलग-अलग Data Structures की आवश्यकता होती है।
5. Data Structure का चयन problem की **nature (प्रकृति – nature)** पर निर्भर करता है।
6. Efficient Data Structure memory space की बचत (Optimization – space optimization) करता है।
7. Time complexity और space complexity दोनों Data Structure से प्रभावित होती हैं।
8. Data Structure program को **simple, readable** और **maintainable** बनाता है।
9. Static और Dynamic memory allocation की concept Data Structure से जुड़ी होती है।
10. Data Structure के बिना large data को handle करना कठिन हो जाता है।
11. Data Structure और Algorithm एक-दूसरे के **पूरक (Complementary – complementary)** होते हैं।
12. सभी advanced topics जैसे Stack, Queue, Tree, Graph की foundation यही है।

### 3 Types / Categories (प्रकार – Definition + Example)

#### ◆ (A) Primitive Data Structures

##### Definition:

वे basic data types जिन्हें directly programming language support करती हैं।

##### Examples:

- int
- float
- char
- boolean

---

#### ◆ (B) Non-Primitive Data Structures

##### Definition:

वे data structures जो primitive data से मिलकर बनते हैं और complex data को store करते हैं।

##### Examples:

- Array
- Linked List
- Stack
- Queue
- Tree
- Graph

---

#### ◆ (C) Static Data Structures

##### Definition:

वे data structures जिनका size fixed होता है और compile time पर define किया जाता है।

##### Example:

- Array

---

#### ◆ (D) Dynamic Data Structures

##### Definition:

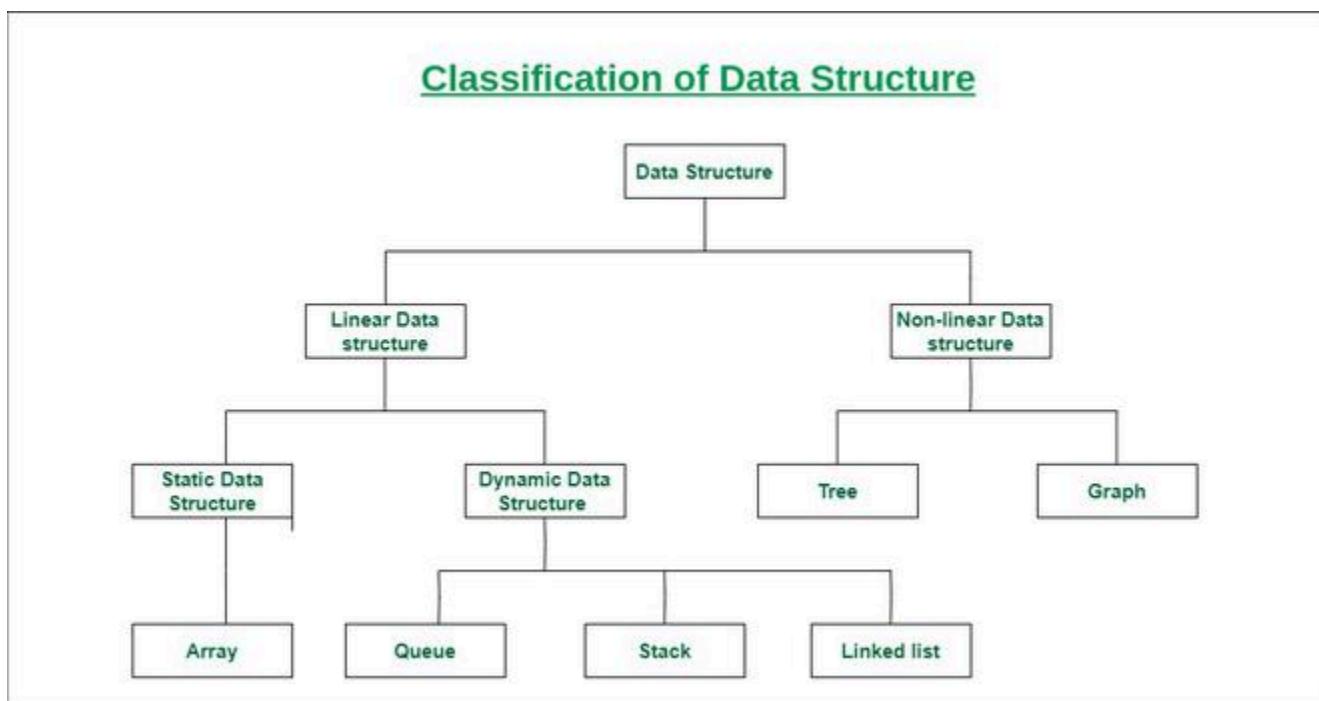
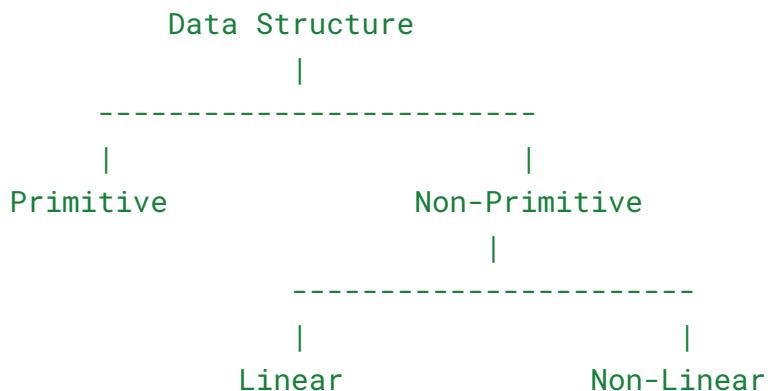
वे data structures जिनका size runtime पर change हो सकता है।

##### Example:

- Linked List

## 4 Diagram Points (Exam में Diagram कैसे बनाएँ)

☞ Diagram: Classification of Data Structures



☞ Diagram tips:

- Boxes साफ और समान आकार के
- Straight lines और arrows
- Heading लिखना मत भूलना

## 5 Examples (2–3 Real Examples)

1. Student Record Management System
2. Bank Account Data Storage
3. Employee Payroll System

---

## 6 Advantages / Features (लाभ / विशेषताएँ)

1. Data को व्यवस्थित रूप में store करता है।
  2. Data processing fast होती है।
  3. Memory utilization बेहतर होता है।
  4. Program efficiency बढ़ती है।
  5. Error handling आसान हो जाती है।
- 

## 7 Disadvantages / Limitations (सीमाएँ)

1. Incorrect Data Structure चयन performance घटा सकता है।
  2. Complex Data Structures समझने में कठिन हो सकते हैं।
  3. Implementation में समय अधिक लग सकता है।
- 

## 8 Applications / Uses (उपयोग)

1. Database Management System
  2. Operating System
  3. Compiler Design
  4. Artificial Intelligence
  5. Web Applications
- 

## 9 Extra Important Information (Topper Line – Exam में लिखने योग्य)

“Data Structure provides an efficient way to organize, manage and process data in computer systems, forming the backbone of efficient programming.”

---

### ✓ Exam Tip (Topper Trick)

अगर question सिर्फ “Introduction to Data Structure” हो:

👉 Definition + 6–8 concept points + diagram + 1 example = Full Marks ✓

# BASIC TERMINOLOGY OF DATA STRUCTURES

(Unit-1 : Introduction to Data Structures)

## 1 Definition (परिभाषा – point-wise)

1. **Basic Terminology of Data Structures** वे मूल शब्द (Basic Terms) हैं जो data के organization, storage और processing को समझने के लिए उपयोग किए जाते हैं।
2. ये terminologies data और information के बीच स्पष्ट अंतर बताती हैं।
3. Data Structure के advanced concepts को समझने के लिए basic terminology का ज्ञान आवश्यक होता है।

## 2 Concept / Explanation (विस्तार से – 10–12 बिंदु)

1. Computer system में data को सीधे उपयोग नहीं किया जा सकता, इसलिए उसे structured form में व्यवस्थित किया जाता है।
2. Data Structure से संबंधित terminologies data की **nature** (प्रकृति – nature) और **behavior** (व्यवहार – behavior) को समझाती हैं।
3. Data raw facts होते हैं, जबकि information processed data होती है।
4. Data items को logical units में divide किया जाता है जैसे field, record और file।
5. Data Type यह निर्धारित करता है कि data memory में कितनी space लेगा और कैसे process होगा।
6. Data items के बीच संबंध (Relationship – relationship) data structure द्वारा दर्शाया जाता है।
7. Primitive और Non-Primitive data structures data की complexity को define करते हैं।
8. Static और Dynamic data structures memory allocation के तरीके बताते हैं।
9. Proper terminology program की readability और clarity बढ़ाती है।
10. Data Structure और Algorithm एक-दूसरे पर निर्भर (Dependent – dependent) होते हैं।
11. Basic terminology clear होने से error कम होते हैं।
12. ये terminologies सभी data structures जैसे Stack, Queue, Tree और Graph की foundation होती हैं।

### **3 Types / Categories (Basic Terminologies – Definition + Example)**

नीचे सबसे important basic terms दिए गए हैं ↴

---

#### **◆ (A) Data**

**Definition:**

**Data** कच्चे तथ्य (Raw Facts – raw facts) होते हैं जिनका स्वयं में कोई अर्थ नहीं होता।

**Example:**

25, "Aman", 78.5

---

#### **◆ (B) Information**

**Definition:**

Processed data को **Information** कहते हैं, जो meaningful और useful होती है।

**Example:**

"Aman scored 78.5 marks"

---

#### **◆ (C) Data Item**

**Definition:**

Data का एक single unit **Data Item** कहलाता है।

**Example:**

Roll Number = 101

---

#### **◆ (D) Field**

**Definition:**

Record का सबसे छोटा भाग **Field** कहलाता है।

**Example:**

Name, Age, Marks

---

#### **◆ (E) Record**

**Definition:**

Related fields के collection को **Record** कहते हैं।

**Example:**

(Student Name, Roll No, Marks)

- ◆ **(F) File**

**Definition:**

Similar records के collection को **File** कहते हैं।

**Example:**

Student File, Employee File

---

- ◆ **(G) Data Type**

**Definition:**

Data की प्रकृति (Nature – nature) को define करने वाले प्रकार को **Data Type** कहते हैं।

**Examples:**

int, float, char, boolean

---

- ◆ **(H) Data Structure**

**Definition:**

Data को memory में व्यवस्थित और संरचित रूप में store करने की तकनीक को **Data Structure** कहते हैं।

**Examples:**

Array, Stack, Queue

---

- ◆ **(I) Primitive Data Structure**

**Definition:**

वे basic data structures जिन्हें programming language directly support करती हैं।

**Examples:**

int, float, char

---

- ◆ **(J) Non-Primitive Data Structure**

**Definition:**

वे complex data structures जो primitive data से मिलकर बनते हैं।

**Examples:**

Array, Linked List, Stack, Tree

---

- ◆ **(K) Static Data Structure**

**Definition:**

जिस data structure का size fixed होता है, उसे **Static Data Structure** कहते हैं।

**Example:**

Array

---

## ◆ (L) Dynamic Data Structure

### Definition:

जिस data structure का size runtime पर change हो सकता है, उसे **Dynamic Data Structure** कहते हैं।

### Example:

Linked List

---

## 4 Diagram Points (Exam में Diagram कैसे बनाएँ)

### Diagram: Organization of Data

```
Data
↓
Field
↓
Record
↓
File
↓
Data Structure
```

### Diagram tips:

- Boxes समान आकार में बनाओ
- Arrows straight रखो
- Side में example लिख सकते हो

---

## 5 Examples (Exam-ready)

1. Student Information System
2. Bank Record Management
3. Library Management System

---

## 6 Advantages / Features (लाभ / विशेषताएँ)

1. Data organization आसान होता है।
2. Data access तेज हो जाता है।
3. Program efficiency बढ़ती है।
4. Memory utilization बेहतर होता है।

---

## 7 Disadvantages / Limitations (सीमाएँ)

1. Terminology clear न होने पर गलत data structure चुन सकते हैं।
2. Complex terms beginners के लिए कठिन हो सकते हैं।
3. गलत understanding से program errors बढ़ सकते हैं।

---

## 8 Applications / Uses (उपयोग)

1. Database systems
2. Operating systems
3. Software development
4. Data analysis applications

---

## 9 Extra Important Information (Topper Line)

“Basic terminology of data structures provides the foundation for understanding how data is organized, stored and processed efficiently in computer programs.”

---

### Exam Tip (Topper Trick)

अगर question आए “Explain basic terminology of data structure”

👉 6–8 terms + diagram + 1 example = Full Marks ✓

---

## CLASSIFICATION OF DATA STRUCTURES

(Unit–1 : Introduction to Data Structures)

---

### 1 Definition (परिभाषा – point-wise)

1. **Classification of Data Structures** वह प्रक्रिया है जिसके द्वारा data structures को उनकी **प्रकृति (Nature – nature)**, **आकार (Size – size)** और **संगठन (Organization – organization)** के आधार पर वर्गीकृत किया जाता है।
2. यह classification data storage और processing को **सरल (Simple – simple)** और **कुशल (Efficient – efficient)** बनाती है।
3. सही classification program की performance और memory utilization सुधारती है।

## 2 Concept / Explanation (विस्तार से – 10–12 बिंदु)

1. Data Structures को classify करने का उद्देश्य data को logical तरीके से organize करना है।
  2. Classification से यह तय होता है कि data memory में कैसे store और कैसे access किया जाएगा।
  3. Different problems के लिए अलग प्रकार के data structures उपयुक्त होते हैं।
  4. Classification program design को आसान बनाती है।
  5. Linear और Non-Linear classification data elements के arrangement को दर्शाती है।
  6. Primitive और Non-Primitive classification data की complexity बताती है।
  7. Static और Dynamic classification memory allocation behavior को समझाती है।
  8. Proper classification से searching, insertion और deletion operations efficient होते हैं।
  9. Classification algorithm selection में भी सहायता करती है।
  10. Large data handling में classification बहुत महत्वपूर्ण होती है।
  11. सभी advanced data structures किसी न किसी category में आते हैं।
  12. Data Structure classification system performance और scalability बढ़ाती है।
- 

## 3 Types / Categories (प्रकार – Definition + Points + Examples)

### ◆ (A) Primitive Data Structures

#### Definition:

वे basic data structures जिन्हें programming language directly support करती हैं।

#### Points:

1. Simple और basic form।
2. Single value store करते हैं।
3. Fixed memory size।

#### Examples:

int, float, char, boolean

### ◆ (B) Non-Primitive Data Structures

#### Definition:

वे data structures जो primitive data से मिलकर बनते हैं और complex data store करते हैं।

**Points:**

1. Multiple values store कर सकते हैं।
2. Logical relationship maintain करते हैं।
3. Advanced data handling के लिए उपयोगी।

**Examples:**

Array, Linked List, Stack, Queue, Tree, Graph

---

**◆ (C) Linear Data Structures****Definition:**

वे data structures जिनमें data elements **linear sequence** में arranged होते हैं।

**Points:**

1. Single level structure।
2. Sequential traversal possible।
3. Implementation आसान।

**Examples:**

Array, Stack, Queue, Linked List

---

**◆ (D) Non-Linear Data Structures****Definition:**

वे data structures जिनमें data elements **hierarchical या network form** में arranged होते हैं।

**Points:**

1. Multiple level structure।
2. Complex relationships possible।
3. Traversal complex होता है।

**Examples:**

Tree, Graph

---

**◆ (E) Static Data Structures****Definition:**

वे data structures जिनका size compile time पर fixed होता है।

**Points:**

1. Memory allocation fixed।
2. Fast access।
3. Memory wastage possible।

**Example:**

Array

## ◆ (F) Dynamic Data Structures

### Definition:

वे data structures जिनका size runtime पर change हो सकता है।

### Points:

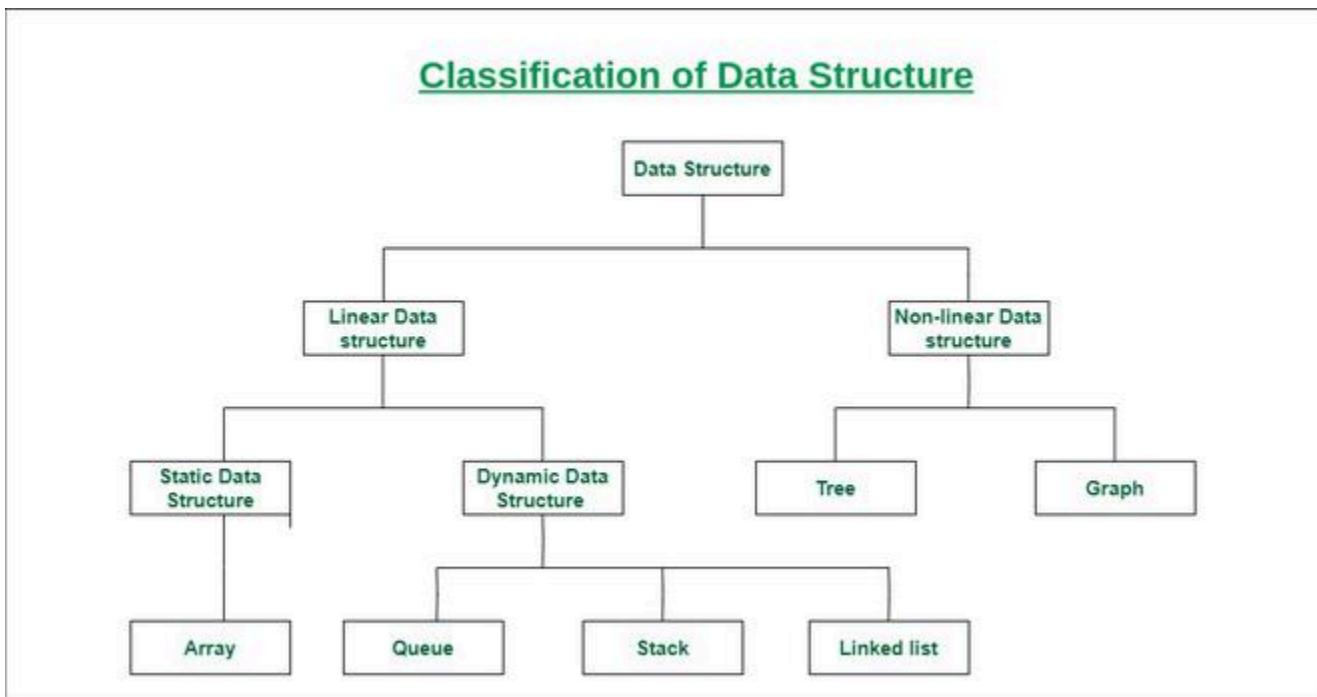
1. Memory efficient!
2. Flexible size!
3. Implementation complex!

### Examples:

Linked List, Tree, Graph

## 4 Diagram Points (Exam में Diagram कैसे बनाएँ)

### ☞ Diagram: Classification of Data Structures



### ☞ Diagram Tips:

- Heading जरूर लिखो
- Boxes समान आकार के
- Lines straight और साफ़

## 5 Examples (2–3 Exam-ready)

1. Student list → Array (Linear)
2. Browser history → Stack

- 
- 3. Organization hierarchy → Tree

---

## 6 Advantages / Features (लाभ / विशेषताएँ)

- 1. Data organization आसान हो जाता है।
- 2. Program design simple बनता है।
- 3. Efficient data access संभव होता है।
- 4. Memory utilization बेहतर होता है।

---

## 7 Disadvantages / Limitations (सीमाएँ)

- 1. गलत classification से performance घट सकती है।
- 2. Dynamic structures implement करना कठिन हो सकता है।
- 3. Non-linear traversal complex होता है।

---

## 8 Applications / Uses (उपयोग)

- 1. Database management systems
- 2. Operating systems
- 3. Compiler design
- 4. Artificial Intelligence
- 5. Network routing

---

## 9 Extra Important Information (Topper Line)

“Classification of data structures helps in selecting the most suitable structure for efficient data storage, processing and problem solving.”

---

### ✓ Exam Tip (Topper Trick)

अगर question आए “**Explain classification of data structures**”

👉 Diagram + 4–5 types with examples = **Full Marks** ✓

# OPERATIONS ON DATA STRUCTURES

(Unit-1 : Introduction to Data Structures)

## 1 Definition (परिभाषा – point-wise)

1. Operations on Data Structures वे क्रियाएँ (Operations) हैं जिनके द्वारा data structure में data को store, access, modify और manage किया जाता है।
2. ये operations data की सुरक्षा (Safety – safety), दक्षता (Efficiency – efficiency) और संगति (Consistency – consistency) बनाए रखने में सहायक होते हैं।
3. Data Structures पर किए जाने वाले सभी operations program की performance को प्रभावित करते हैं।

## 2 Concept / Explanation (विस्तार से – 10–12 बिंदु)

1. Data Structures केवल data को store करने के लिए नहीं, बल्कि उस data पर operations करने के लिए बनाए जाते हैं।
2. Different data structures में operations का तरीका अलग-अलग हो सकता है।
3. Operations data की insertion, deletion और retrieval को नियंत्रित करते हैं।
4. Efficient operations program की execution time कम करते हैं।
5. Operations data integrity (डेटा की अखंडता – integrity) बनाए रखते हैं।
6. Linear और Non-Linear data structures में operations की complexity अलग होती है।
7. Searching operation data को locate करने में सहायता करता है।
8. Traversing operation data elements को sequentially access करता है।
9. Sorting operation data को ordered form में arrange करता है।
10. Merging operation multiple data structures को combine करता है।
11. Operations algorithm के design पर निर्भर करते हैं।
12. Proper operation selection memory utilization और performance सुधारता है।

### 3 Types / Categories (मुख्य Operations – Definition + Points + Example)

नीचे सबसे महत्वपूर्ण operations दिए गए हैं 👇

#### ◆ (A) Traversing

##### Definition:

Data Structure के सभी elements को एक-एक करके access करने की प्रक्रिया को Traversing कहते हैं।

##### Points:

1. Read-only operation।
2. Data elements sequentially visit होते हैं।
3. Linear structures में आसान।

##### Example:

Array के सभी elements print करना।

#### ◆ (B) Insertion

##### Definition:

Data Structure में नया element add करने की प्रक्रिया को Insertion कहते हैं।

##### Points:

1. Beginning, middle या end में insertion।
2. Memory space आवश्यक।
3. Wrong insertion से data order बिगड़ सकता है।

##### Example:

Array में नया element add करना।

#### ◆ (C) Deletion

##### Definition:

Data Structure से किसी existing element को remove करने की प्रक्रिया को Deletion कहते हैं।

##### Points:

1. Memory free होती है।
2. Correct index/position आवश्यक।
3. Improper deletion से data loss हो सकता है।

##### Example:

List से student record हटाना।

---

#### ◆ (D) Searching

##### **Definition:**

Data Structure में किसी specific element को **find** करने की प्रक्रिया को **Searching** कहते हैं।

##### **Points:**

1. Linear Search और Binary Search common।
2. Performance data structure पर depend करती है।
3. Sorted data में searching fast होती है।

##### **Example:**

Roll number search करना।

---

#### ◆ (E) Sorting

##### **Definition:**

Data elements को **ascending या descending order** में arrange करने की प्रक्रिया को **Sorting** कहते हैं।

##### **Points:**

1. Searching को fast बनाता है।
2. Bubble, Insertion, Quick sort common।
3. Large data handling में important।

##### **Example:**

Marks को ascending order में arrange करना।

---

#### ◆ (F) Merging

##### **Definition:**

दो या अधिक data structures को **एक साथ combine** करने की प्रक्रिया को **Merging** कहते हैं।

##### **Points:**

1. Similar type structures required।
2. Sorted merging fast होता है।
3. Used in file processing।

##### **Example:**

Two sorted arrays को merge करना।

---

## 4 Diagram Points (Exam में Diagram कैसे बनाएँ)

### Diagram: Operations on Data Structures



### Diagram Tips:

- Center में “Data Structure”
- नीचे operations को boxes में
- Straight lines और clear labels

## 5 Examples (2–3 Exam-ready)

1. Student List → Insertion & Deletion
2. Library System → Searching
3. Result Processing → Sorting

## 6 Advantages / Features (लाभ / विशेषताएँ)

1. Data management efficient बनता है।
2. Program performance improve होती है।
3. Data integrity बनी रहती है।
4. Large data processing संभव होता है।

## 7 Disadvantages / Limitations (सीमाएँ)

1. Complex operations implementation कठिन।
2. Wrong operation selection से performance घट सकती है।
3. Memory overhead बढ़ सकता है।

## 8 Applications / Uses (उपयोग)

1. Database management systems
2. Operating systems
3. File processing systems
4. Scientific applications

## 9 Extra Important Information (Topper Line)

**“Operations on data structures define how data is accessed, modified and managed efficiently, directly affecting program performance.”**

---

### ✓ Exam Tip (Topper Trick)

अगर question आए “**Explain operations on data structures**”

👉 Definition + 5 operations + diagram + example = **Full Marks** ✓