In this chapter

* What is data structure?
* Why we need Data Structure and Algorithms?
* Data Types and Data Structures

***What is Data Structure?***

Data structures are methods used to organize and store data in a computer's memory or on disk. They define how data is arranged, accessed, and manipulated efficiently. Examples include arrays, linked lists, stacks, queues, trees, and graphs, each designed for specific types of operations and performance characteristics. Understanding data structures is fundamental for efficient algorithm design and software development.

1. Domian (D): The set of possible values that the data can take.
2. Function (F): The operations that can be performed on the data objects within (D).
3. Axioms (A): The rules that govern how these operations are implemented.

A data structure D is defined by the triplet (D F A), where D is the set of data objects, F is the set of operations/functions on these objects, and A specifies the rules for implementing these functions.

***Why we need Data Structure and Algorithms?***

1. **Efficiency**: They enable efficient storage, retrieval, and manipulation of data. Efficient algorithms ensure that programs run quickly and use minimal resources.
2. **Problem Solving**: They provide standardized methods to solve common programming problems, such as searching, sorting, and organizing data.
3. **Optimization**: Properly chosen data structures and algorithms optimize program performance, making operations faster and more scalable.
4. **Resource Management**: They help manage memory and other resources effectively, preventing wastage and improving overall program efficiency.
5. **Complexity Management**: They allow programmers to handle complex problems by breaking them down into simpler, manageable tasks using structured approaches.
6. **Foundation of Programming**: They form the core concepts taught in computer science courses, providing a solid foundation for understanding and developing software applications.

In summary, data structures and algorithms are crucial tools that enable programmers to write efficient, scalable, and maintainable software solutions to a wide range of problems.

***Data Types in Java***

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **Description** | **Size** | **Range** |
| byte | 8-bit signed integer | 1 byte | -128 to 127 |
| short | 16-bit signed integer | 2 bytes | -32,768 to 32,767 |
| int | 32-bit signed integer | 4 bytes | -2^31 to 2^31 - 1 |
| long | 64-bit signed integer | 8 bytes | -2^63 to 2^63 - 1 |
| float | 32-bit IEEE 754 floating point | 4 bytes | Approximately ±3.40282347E+38F |
| double | 64-bit IEEE 754 floating point | 8 bytes | Approximately ±1.79769313486231570E+308 |
| boolean | Represents true or false | 1 bit | true or false |
| char | 16-bit Unicode character | 2 bytes | '\u0000' (0) to '\uffff' (65,535) |

***Data Structure***

* Arrays
* Linked List
* Stacks
* Queue
* Trees
* Graph
* Table
* Set

***Algorithms***

* + Insertion
  + Searching
  + Sorting
  + Iterate