**DATA STRUCTURES AND ALGORITHMS IN JAVA**

What are Data Structures in Java?

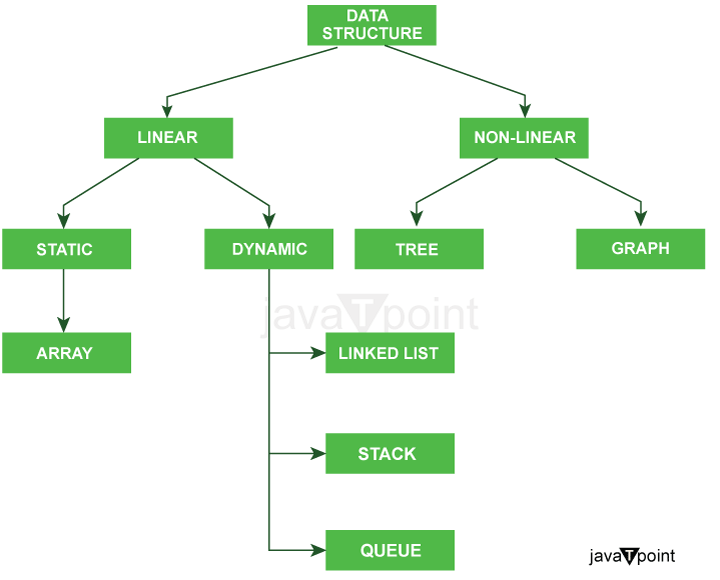
The way that data is organized and stored in a computer program's memory relies closely on Java records structures. The Java well-known library includes a significant type of built-in statistics structures. A few of the records systems that permit programmers short and simple ways to save and arrange data include connected lists, stacks, queues, and arrays. Developers can quickly perform operations like insertion, deletion, searching, and sorting because they provide a range of mechanisms for getting Access to, altering, and managing data. Java programmers can reduce memory use and considerably boost the overall efficiency of their programs by using these data structures.

Types of Data Structures in Java

The list of data structures in Java listed below

1. Arrays
2. Array-List
3. LinkedList
4. Stack
5. Queue
6. HashMap
7. HashSet
8. Tree-Set
9. Tree-Map
10. Graph
11. Tree

The below diagram clearly explains the types of Data Structures in Java very clearly.



Further classification of types of Data Structures:

**There are two types of Data Structures: -**

1. Primitive Data Structures
2. Non-primitive Data Structures

**1) Primitive Data Structures:** Also known as primitive data types, these are basic built-in data types in Java. They include:

* **Byte:** Stores whole numbers from -128 to 127.
* **short:** Stores whole numbers from -32,768 to 32,767.
* **int:** Stores whole numbers from -2,147,483,648 to 2,147,483,647.
* **float:** Stores floating-point numbers with single precision.
* **char:** Stores individual characters.
* **boolean:** Stores true or false values.
* **long:** Stores large whole numbers.
* **Double:** Stores floating-factor numbers with double precision.

**2) Non-primitive Data Structures:** Non-primitive records structures are more complex and are composed of primitive information sorts. They may be, in addition, categorized into two sorts:

1. **Linear Data Structures:** In linear data structures, the elements are arranged linearly or sequentially. Examples include:
   * **Arrays:** A group of identically-typed elements placed in an array according to a predetermined arrangement.
   * **Stacks:** A Last-In-First-Out (LIFO) structure in which only the topmost items may be added or removed.
   * **Queues:** First-In-First-Out (FIFO) structures are utilized in queues, where items are inserted on the returned and taken out on the front.
   * **Linked List:** A related list comprises a collection of gadgets referred to as nodes, each of which has a reference to the node after it and statistics inside it.
2. **Non-linear Data Structures:** In non-linear data structures, the elements are arranged in a non-sequential manner. Examples include:
   * **Trees:** Trees are a type of node-based hierarchical structure, with a root node at the top and child nodes branching out of it. Examples include red-black trees, AVL trees, binary search trees, and binary trees.
   * **Graphs:** A set of nodes linked by using edges, wherein nodes may have any quantity of connections. Graphs are used to symbolize complex relationships among items.
   * **Heap:** A specialized tree-based structure in which every determined node has a value more or smaller than its kids, relying on whether or not it is a max heap or a min heap.
   * **Hash:** Data structures that use a hash function to map keys to values. Examples consist of hash sets and hash maps, which provide green retrieval and storage of statistics based on precise keys.