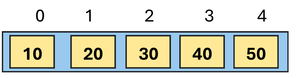
Data Structures

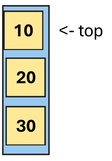
Data structures are structures (ADT s) that provide a specific way of storing and organizing data.

They are basically ADTs that help in storing data

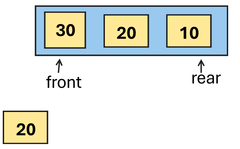
Arrays



Stacks



Queues



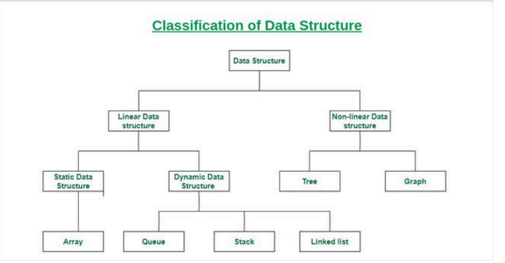
Abstract Data Types

Abstract data types specifies a set of operations and the behavior of these operations, without detailing the underlying implementation or how the data is stored in memory.

an ADT defines what operations can be done — like pushing, popping, or inserting — without revealing how those operations are implemented internally.

```cpp
#include <iostream>
using namespace std;

class Circle {
private:
    double radius;

public:
    void setRadius(double r) {
        radius = r;
    }

    double getArea() {
        return 3.14159 * radius * radius;
    }
};

int main() {
    Circle c;
    c.setRadius(5.0);
    cout << "Area: " << c.getArea() << endl;
    return 0;
}
```

Classification of data structures



In linear data structure, data elements are arranged sequentially i.e. one after the other.Where as in non linear data structures data elements are not arranged sequentially and more like a hierarchy.

Aspect	Static Data Structure	Dynamic Data Structure
Memory allocation	Memory is allocated at compile-time	Memory is allocated at run-time
Size	Size is fixed and cannot be modified	Size can be modified during runtime
Memory utilization	Memory utilization may be inefficient	Memory utilization is efficient as memory can be reused
Access	Access time is faster as it is fixed	Access time may be slower due to indexing and pointer usage
Examples	Arrays, Stacks, Queues, Trees (with fixed size)	Lists, Trees (with variable size), Hash tables