ADS LAB ASSIGNMENT 1

Name - Khushbu Jain
ROU Number - 23115047

Q. Explain Abstract Data Type (ADT)? Describe the concept of abstraction?

Sol → Definition of ADT (Abstract data type):

- Abstract data type is a theoretical concept used in computer science.
- It defines what a data structure does, not how it does it.
- It encapsulates data and the operations performed on it, hiding its implementation details.
- ADT is independent of any programming language or implementation.

Key Concepts:

i) Abstraction:

- Abstraction is the process of hiding implementation details & showing only the functionality.
- It focuses on "what operations" are possible, not "how they are performed".
- Helps in simplifying complex systems by exposing only relevant features.

ii) Encapsulation:

- Data & operation are bundled together.
- Internal representation is hidden from the user (data integrity is maintained).

iii) Operations in ADT:

Examples of common operations in ADTs:

- Stack : push(), pop()
- Queue : enqueue(), dequeue()
- List : insert(), delete(), search()

Code Example (stack as ADT)

```
#include <iostream>
using namespace std;
class stackADT {
private:
    int *arr;
    int top;
    int size;
public:
    stackADT(int capacity){
      size = capacity;
      arr = new int[size];
      top = -1;
    }
    void push(int x){
       if (top == size-1){
          cout << "stack Overflow" <<endl;
          return;
       }
       arr[top++] = x;

    }
    int peek() {
       if (top == -1) return -1;
       return arr[top];
    }
    void pop() {
       if(top == -1) {
          cout << "Stack Overflow" <<endl;
          return;
       }
       top--;
    }
    bool isEmpty() {
       return top == -1;
    }
};
```

```
int main(){
    stackADT s(5);
    s.push(10);
    s.push(20);
    s.push(30);
    cout<<"Top element: "<<s.peek()<<endl;
    s.pop();
    cout<<"After pop, top element: "<<s.peek()<<endl;
    return 0;
}
```

Output:

```
Top element : 30
After pop, top element: 20
```