DSA SERIES

- Learn Coding

Topic to be Covered today

Memoization

LETS START TODAY'S LECTURE

Fibonacci

```
class Solution {
public:
    unordered_map<int, long long> dp;

    long long fib(int n) {
        if (n == 0) return 0;
        if (n == 1) return 1;

        if (dp.count(n)) return dp[n];  // already computed

        return dp[n] = fib(n - 1) + fib(n - 2);
    }
};
```

Factorial

```
class Solution {
public:
    unordered_map<int, long long> dp; // memoization map

    long long fact(int n) {
        if (n == 0 || n == 1) return 1;

        if (dp.count(n)) return dp[n]; // already computed

        return dp[n] = n * fact(n - 1);
    }
};
```

Staircase(Count ways to reach n)

```
#include <bits/stdc++.h>
using namespace std;

int dp[1000];

int countWays(int n) {
    if (n == 0) return 1;   // 1 way (stay there)
    if (n == 1) return 1;   // 1 way (only 1 step)

    if (dp[n] != -1) return dp[n];  // already solved

    return dp[n] = countWays(n-1) + countWays(n-2);
}

int main() {
    memset(dp, -1, sizeof(dp));
    int n = 4;
    cout << countWays(n);  // Output: 5
}
```

Learn coding

THANK YOU