Session – 10

Recursion

CONTENTS

01

Introduction to Recursion

Introduction to Recursion

- Recursion is applicable to problems which can be split into similar subproblems and we can solve each subproblem to solve the large problem.
- This would see the function being invoked from within itself, making it a recursive function.
- In some cases, recursion is an easy and more comprehensible way to solve a problem.
- Popular uses: Merge Sort, Quick Sort, Binary Search, Depth First Search/Breadth First Search

Introduction to Recursion

How does recursion work?

- The first function call creates an instance of the function/method on the Call Stack. Every time there is a recursive call, this instance gets paused (its state is maintained) and a new instance gets added to the top of the Call Stack.

- When the terminating condition is met and an instance of the method returns, that instance is popped off the Call Stack and the next instance gets the returned value. This instance in turn returns and gets popped and so on. Finally, the first instance on the Call Stack gets the returned value and finishes up the solution, clearing the Call Stack.

02

General Approach to Recursion

General Approach to Recursion

Basic structure

```
void recursiveFunction() {
    if (baseCondition) {
        return;
    }
    // logic
    recursiveFunction(); // recursive call
}
```

03

Key Aspects

Key Aspects

Terminating Condition

- This is the condition that will result in a recursive instance of the method doing a return. Without this, the recursion might go on till there is no space on the Call Stack. It is important that this condition is met for the solution to be complete.

Recursive Invocation Condition

- Based on the goal of the problem, the function gets recursively invoked with reducing input parameters, till the terminating condition is met.

Recursion Depth

- This is something to keep an eye out for. If the number of instances on the Call Stack exceeds the maximum number supported for that language, you will see a “Maximum call stack size exceeded” exception.

Activity 1

- Letter Combination of a Phone Number
- Generate Parentheses
- Next Permutation
- Towers of Hanoi