Recursion Class 1

Prerequisites

- Iteration / Loops
- Functions

Maths

$f(x) = x^2$ — given

$\boxed{f(f(x))} = f(x^2)$

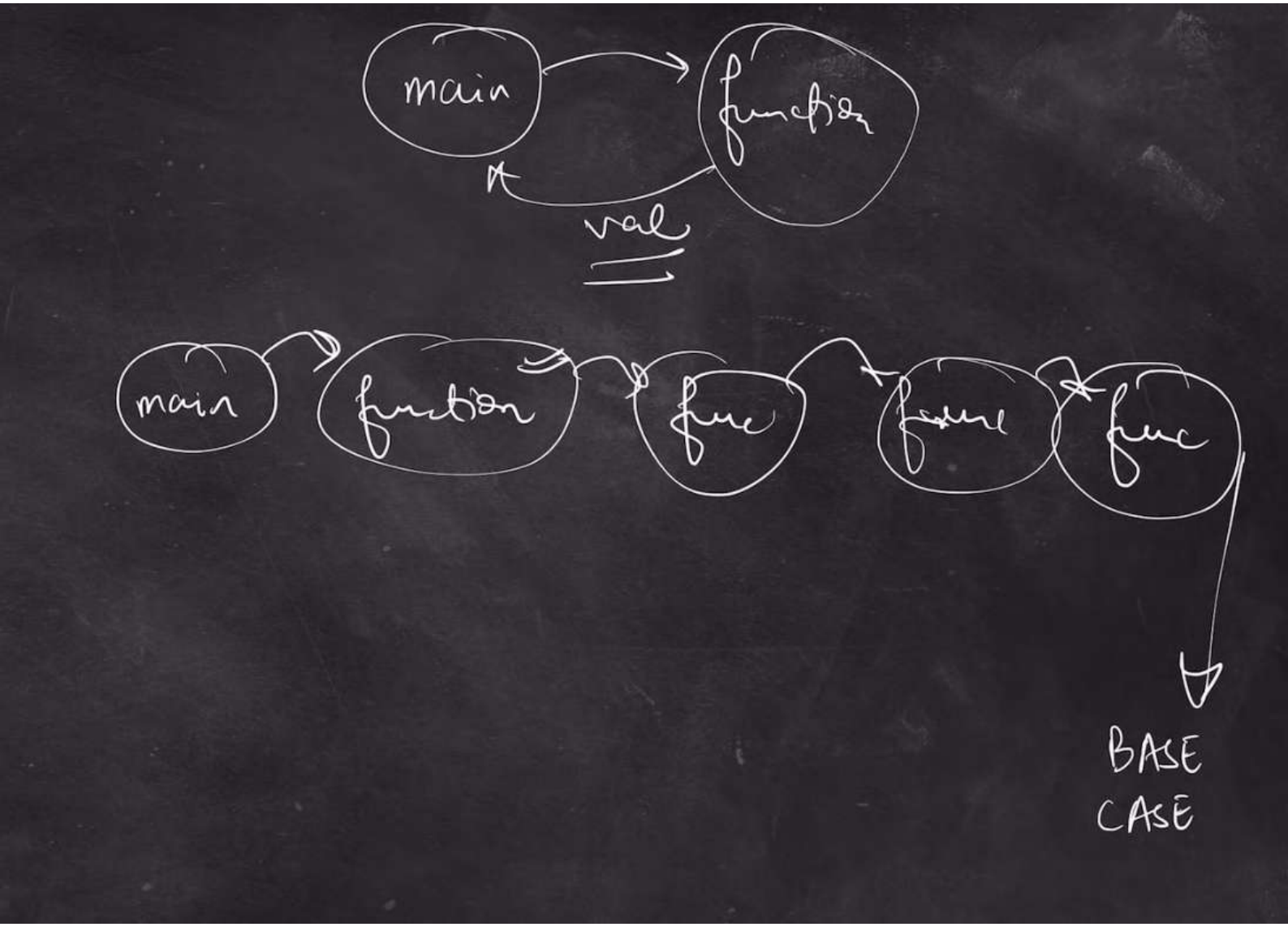
$\longrightarrow$ Recursion

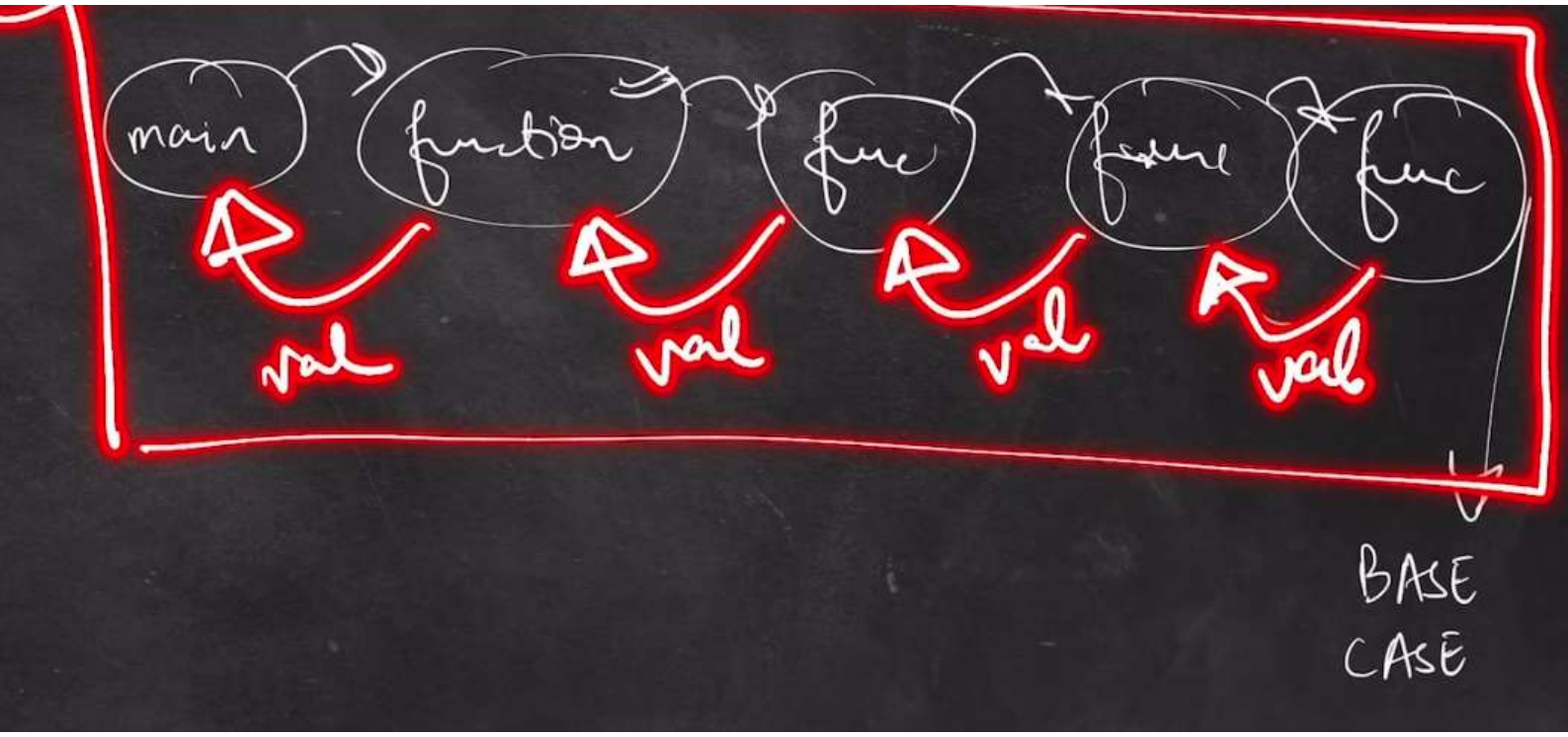
function that calls itself.

$\underline{x = 2}$

$f(x) = 2^2 = \underline{4}$

$f(f(x)) = f(4) = 4^2 = \underline{\underline{16}}$


main
function
val
main
function
func
func
func
BASE
CASE


main
function
func
func
func
val
val
val
val
BASE
CASE

Qs. Print Numbers from 5 to 1

```
for(int i=5 ; i>0 ; i--)
    syso(i);
```

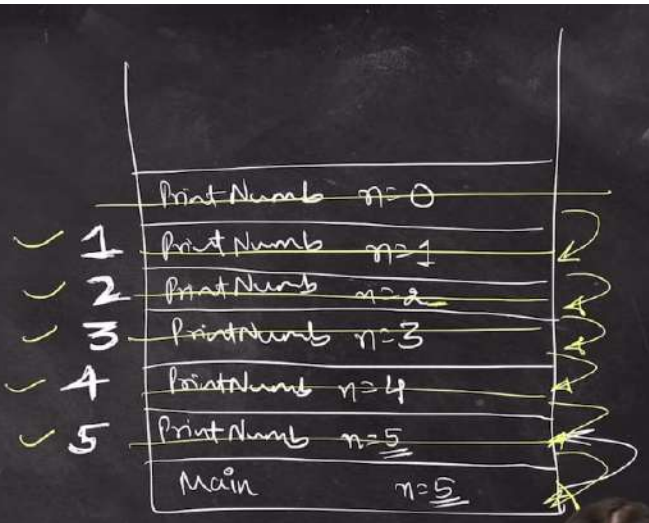
```
public static void printNumb(int n) {
    if(n==0)
        return ; } BASE
    syso(n) ; } print
    printNumb(n-1); } recursion
```

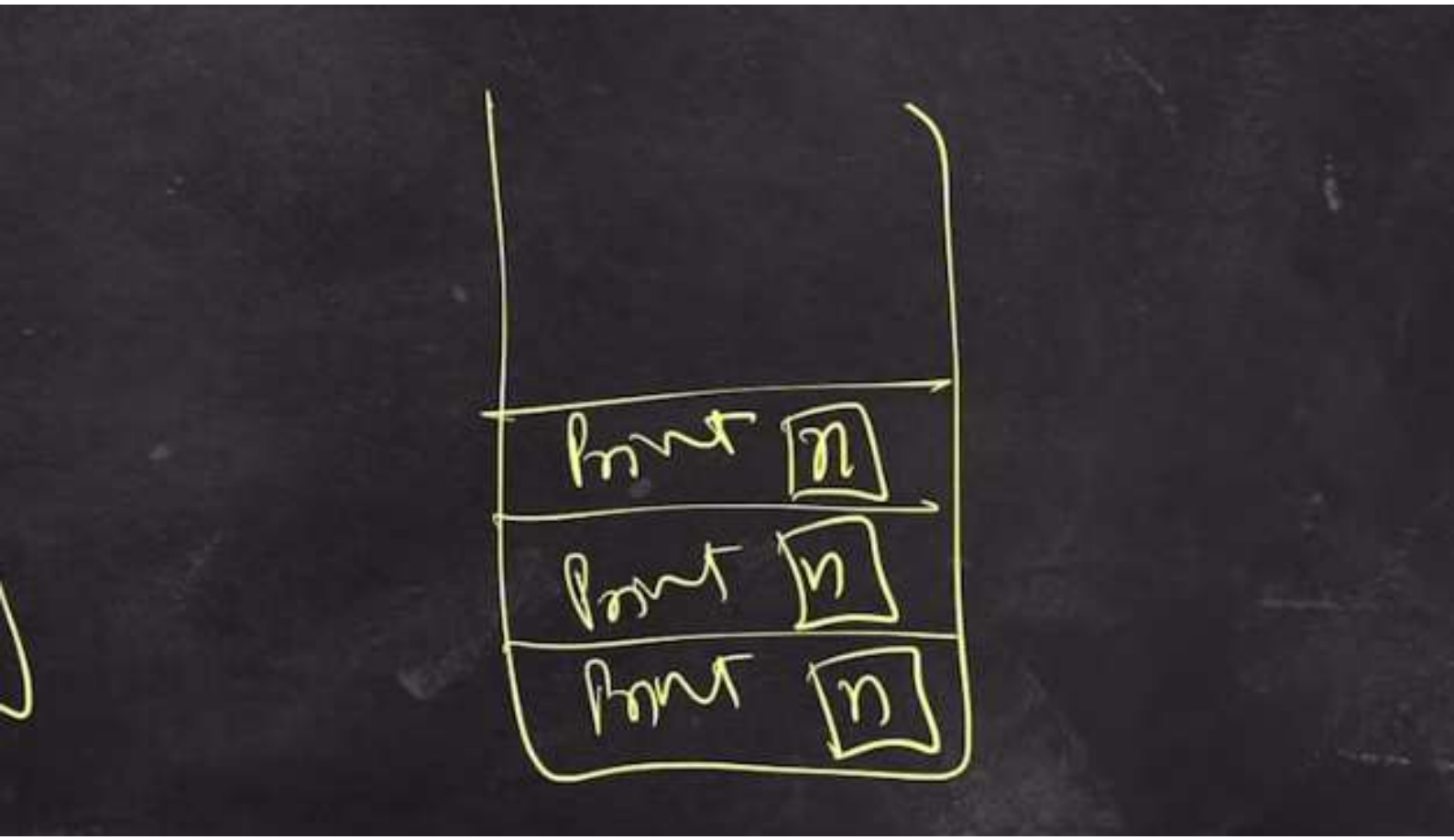
Recursion Class 1

What happens in Memory?

```
public static void printNumb(int n) {
    if(n==0)
        return ; } BASE

    syso(n) ; } print        n=1
    printNumb(n-1); } recursion.
        n=2
```




Print n
Print n
Print n

STACK OVERFLOW

Recursion Class 1

Qs. Print Numbers from 1 to 5

5 → 1

1 → 5

Main

{ 6 Base }

fun(int n)

{

← Base Case

}

```java
public class Recursion1 {
    public static void printNumb(int n) {
        if(n == 6) {
            return;
        }
        System.out.println(n);
        printNumb(n+1);
    }
    Run | Debug
    public static void main(String args[]) {
        int n = 1;
        printNumb(n); // n=1
    }
}
```

Recursion Class 1

Qs. Print sum of first n natural numbers

```java
public class Recursion1 {
    public static void printSum(int i, int n, int sum) {
        if(i == n) {
            sum += i;
            System.out.println(sum);
            return; //?
        }
        sum += i;
        printSum(i+1, n, sum);
    }
    Run | Debug
    public static void main(String args[]) {
        printSum(1, 5, 0);
    }
}
```

Recursion Class 1

Qs. Print Factorial of a number n

```java
public class Recursion1 {
    public static int calcfactorial(int n) {
        if(n == 1 || n == 0) {
            return 1;
        }
        int fact_nm1 = calcfactorial(n-1);
        int fact_n = n * fact_nm1;
        return fact_n;
    }
    Run | Debug
    public static void main(String args[]) {
        int n = 5;
        int ans = calcfactorial(n);
        System.out.println(ans);
    }
}
```

Qs. Print the fibonacci sequence till nth term*

$\rightarrow a \quad b \quad c$

$\downarrow$

$\boxed{c = a + b}$

0 1 1 2 3 5 .....

```java
public class Recursion1 {
    public static void printFib(int a, int b, int n) {
        if(n == 0) {
            return;
        }
        int c = a + b;
        System.out.println(c);
        printFib(b, c, n-1);
    }
    Run | Debug
    public static void main(String args[]) {
        int a = 0, b = 1;
        System.out.println(a);
        System.out.println(b);
        int n = 7;
        printFib(a, b, n-2);
    }
}
```

Qs. Print x^n (stack height = n)

x^n

$x^n = x * x * x * x * \cdots$ n times

① x, n

Recursion Class 1

Qs. Print x^n (stack height = logn)

Qs. Print x^n (stack height = logn)

x^n

$x^{n/2} * x^{n/2} * x$

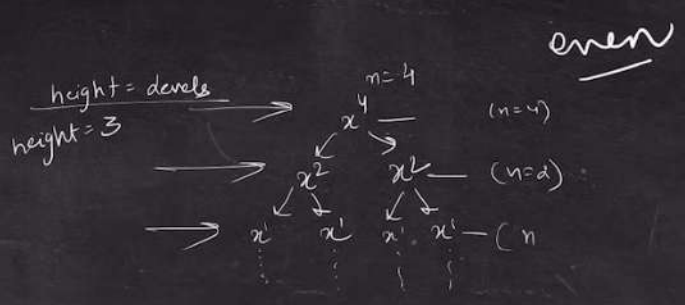
$n \rightarrow$ positive integer

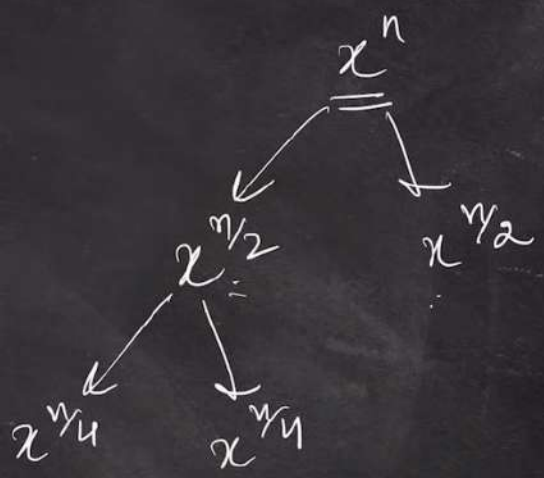
x^n

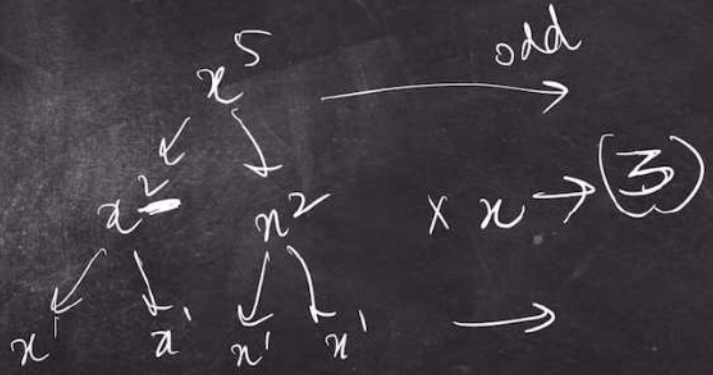
$x^{n/2} * x^{n/2}$

$n \rightarrow$ even power

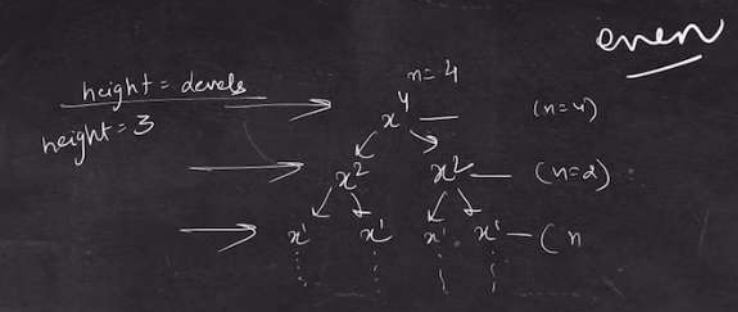
Recursion Class 1

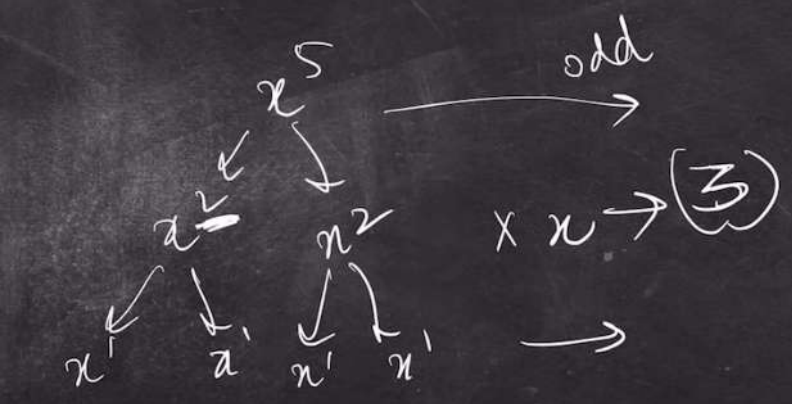


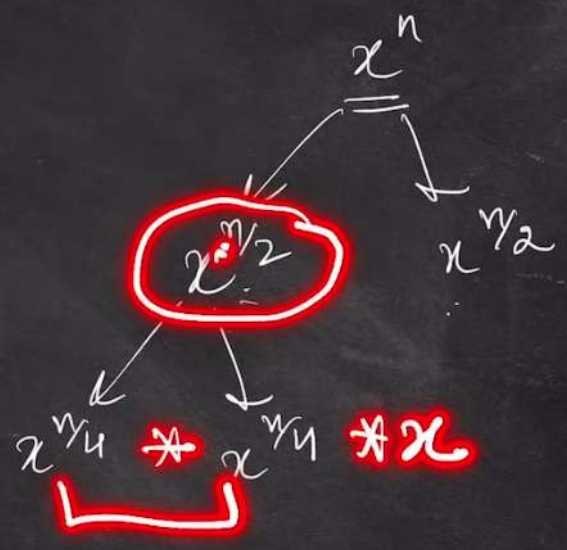




Recursion Class 1







```java
public class Recursion1 {
    public static int calcPower(int x, int n) {
        if(n == 0) { //base case 1
            return 1;
        }
        if(x == 0) { //base case 2
            return 0;
        }
        //if n is even
        if(n % 2 == 0) {
            return calcPower(x, n/2) * calcPower(x, n/2);
        }
        else { // n is odd
            return calcPower(x, n/2) * calcPower(x, n/2) * x;
        }
    }
    Run | Debug
    public static void main(String args[]) {
```