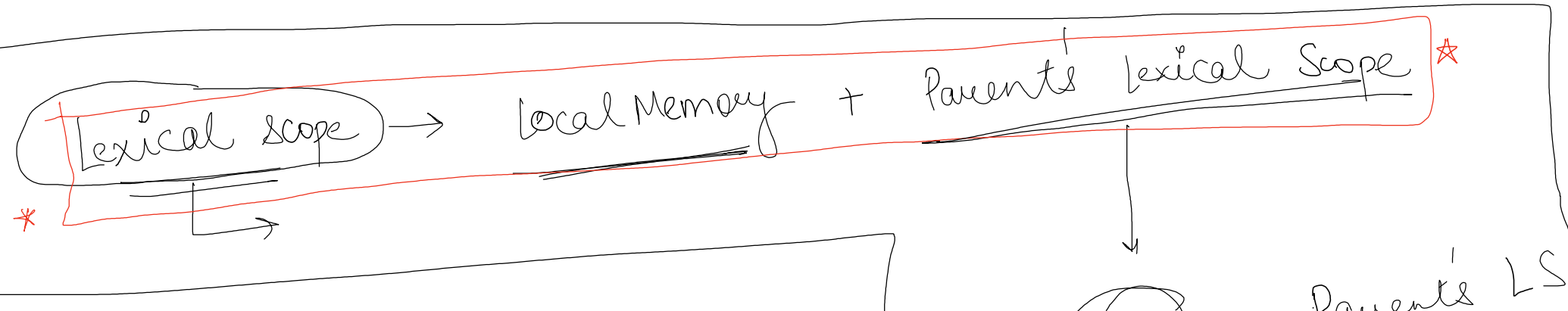
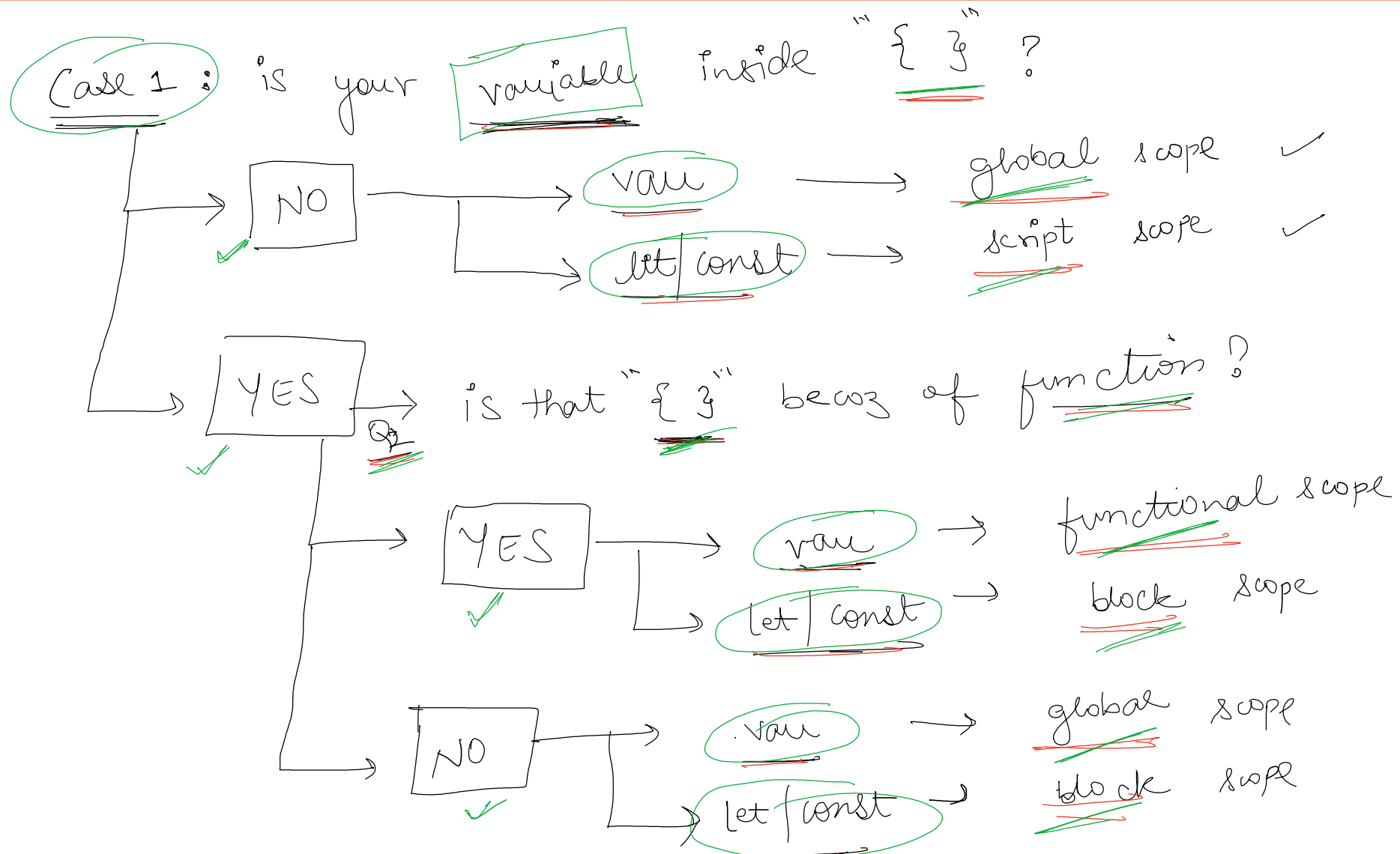


Day-16()

Tuesday, 28 January 2025 7:04 PM

Scope → closure
Questions

Scope

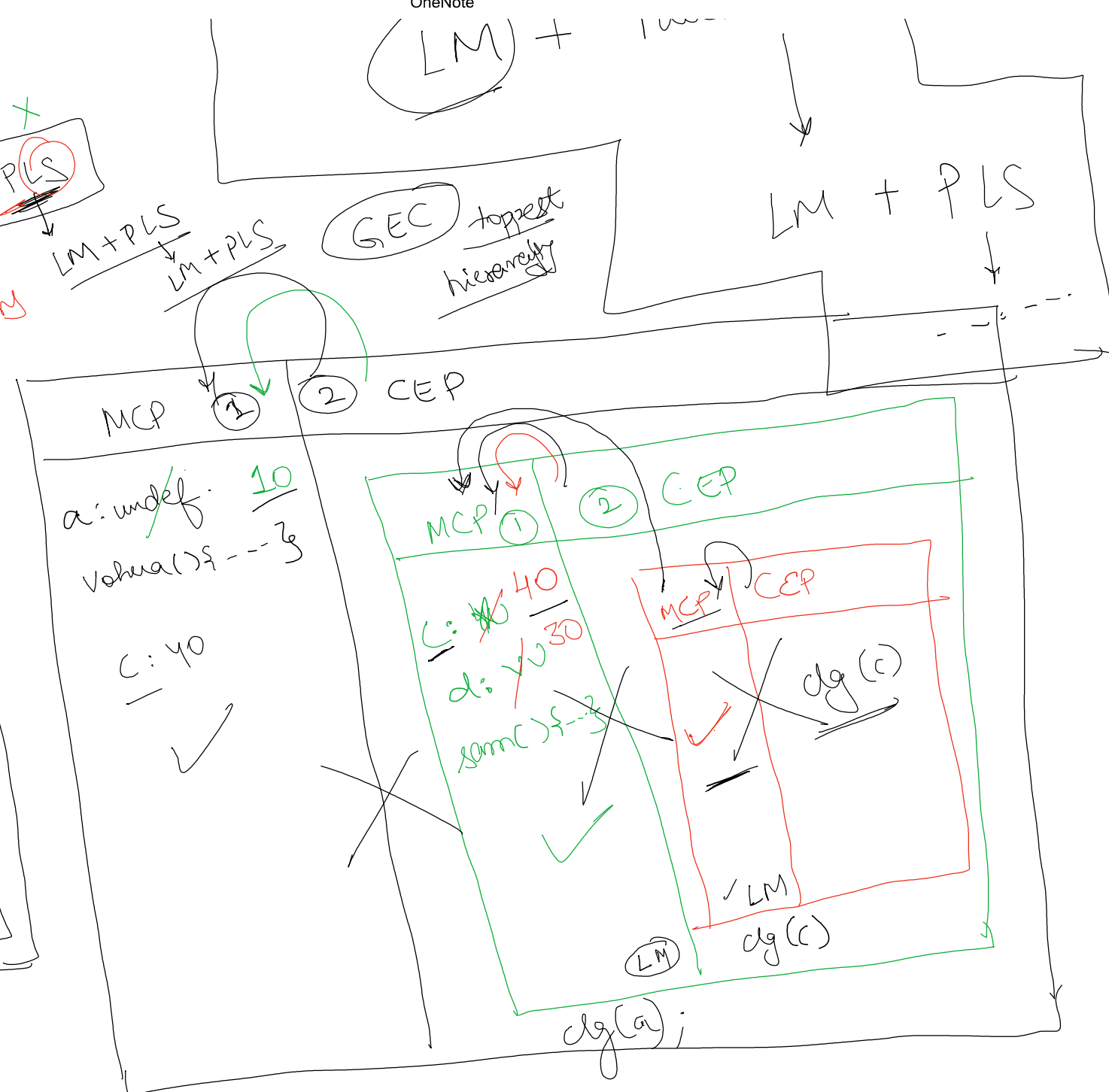
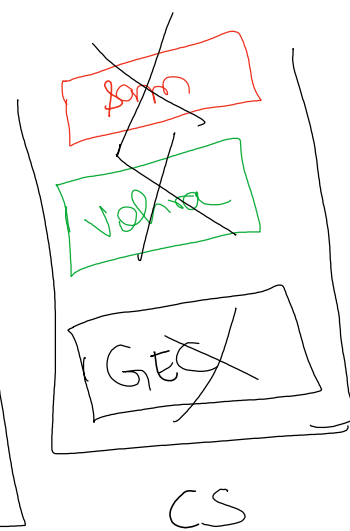
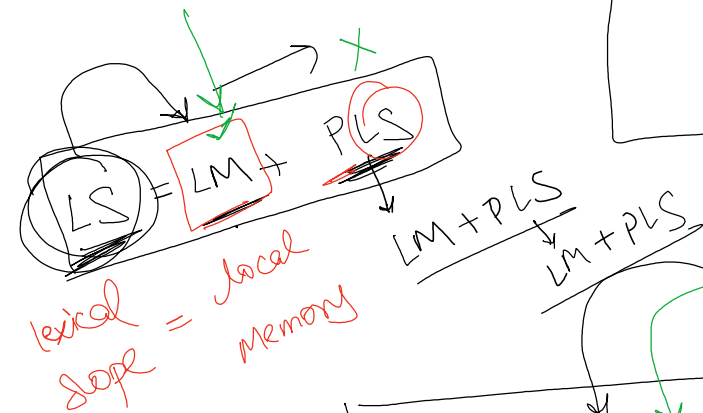
```

1 var a = 10;
2 function vohra(){
3   let c = 40;
4   let d = 30;
5   function sam(){
6     console.log(c);
7   }
8   sam();
9   console.log(c);
10 }
11 vohra();
12 console.log(a);

```

40
40
10

output



```

var a = 10;
let b = 20;

```

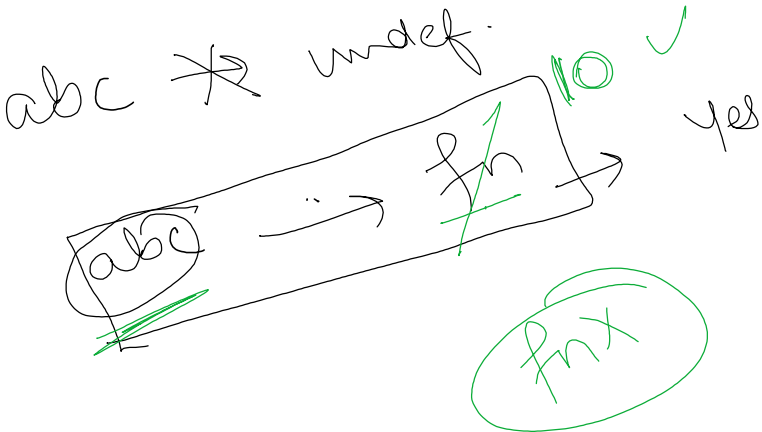
```
function sam(){
  console.log(a);
  var a = 200;
  let b = 500;
  console.log(b);
}
console.log(a);
sam()
console.log(a);
```

MCP

C=1

- ✓ 1 abc() ✓
- ✓ 2 var abc = 10;
- 3 function abc(){
- 4 console.log("hi guyz");
- 5 }
- 6 abc() ✓

Hi guyz
error

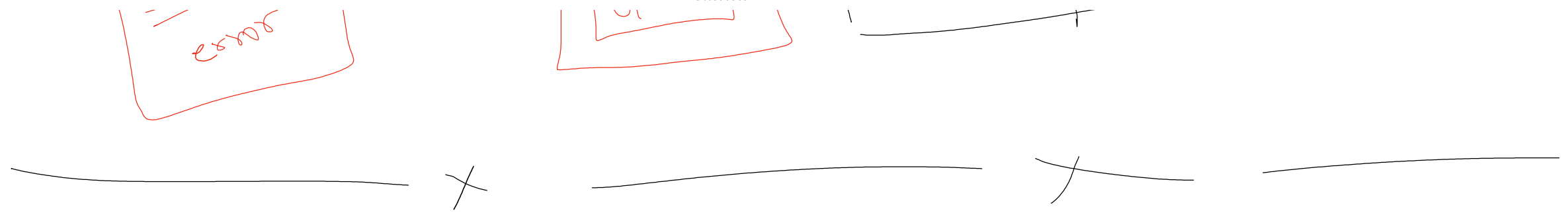


```
abc() ✓
var abc = 10; ✓
function abc(){
  console.log("hi guyz");
}
abc() ✓
```

Hi guyz

15 TC

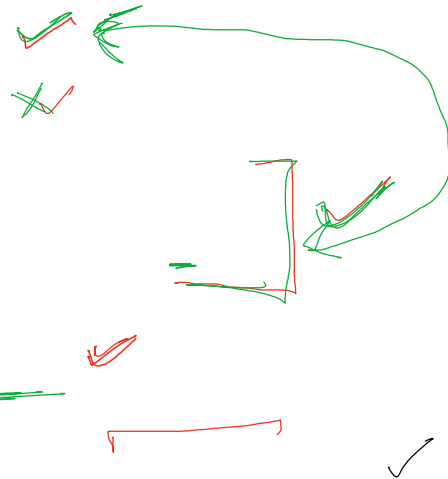
| MCP (1) | CEP (2) |
|---|---------|
| abc → undef function abc() { console.log("hi guyz"); ✓ abc → 10 | |



Closure

Whenever you return a fn, that fn is never returned alone. It is always returned with its lexical dependant environment / scope.

```
function outer(){
  var a = 10;
  let b = 20;
  function inner(){
    console.log(a);
  }
  return inner;
}
```



| MCP ① | ② CEP |
|-----------------|--|
| outer() { ... } | <div> MCP ① ② CEP </div> <div> a: undef. 10 b: v 20 inner() { ? } </div> |
| return: v 10 | |

let returnedVal = outer();
returnedVal()

