**Q1. below line return "bar". Can you explain why?**

**((123 && "foo") && "bar") || null && "baz"**

=>

&& and || is a logical operator which return Boolean values to the variable.

Logical && operator return true when both condition is true otherwise it return false.

Logical || operator always returns true when firtst value return true.

For e.g. (9<3) && (9>3) // it returns false

Properties of && operator is check first value if it is true then it will check second value if second value is also true then it returns right hand value,if the first value is false then it will not check second value it directly returns false.

For e.g. (9<3) || (9>3) //returns true

Properties of || operator is if the first value returns true it will not check second value or condition and return true.

So with help of this theory concept ((123 && "foo") && "bar") || null && "baz" returns bar

because 123 is truthy value and “foo” is also truthy value so it became a true value, while “bar” is also truthy value so the total condition is true and

the first condition is true || oprator will not check second condition.

So operation is perform left to right then it returns bar

**Q2. what are the truthy value and falsy value in JS?**

=>

All the non-primitive values are always truthy values because it contains object & object is a container.

the number except 0 is truthy, true ,var v1= ”string” ,[] this all the truthy values.

Primitive values is falsy values means which contains null, NAN, undefined is falsy.

e.g. var v1 = ”” //blank string is a falsy

var digit = 0 //falsy

var any = null || undefined // falsy

**Q3. What is hoisting in Javascript?**

=>

In JavaScript, variables and functions are declare to the top of the scope before execution is called hoisting So we can use variable and function before declaring them.

e.g a=100;

var b=20;

console.log(a/b);

var a;

**Q4. What is local scope and global scope in javascript?**

=>

The variable declared within a function is a local variable for that function and the variable is declared outside the function scope it is known as global variable.

E.g:

Var a=100; //global variable

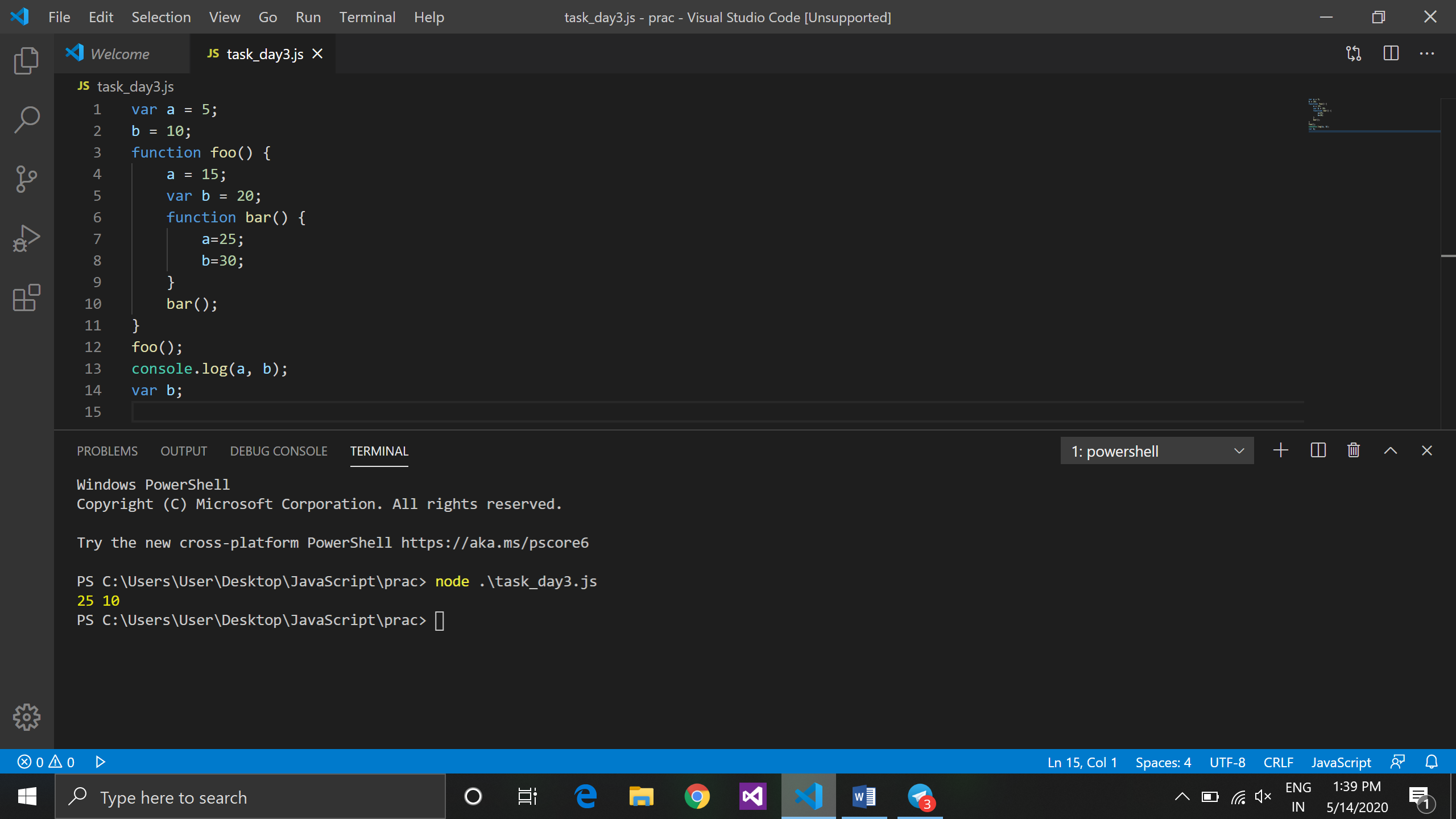
Function check()

{ Var a=10; //local variable

Console.log(a);

}

**Q5. what will be the output of a & b and explain why?;**



1. var a = 5;
2. b = 10;
3. function foo()
4. {
5. a = 15;
6. var b = 20;
7. function bar() {
8. a=25;
9. b=30;
10. }
11. bar();
12. }
13. foo();
14. console.log(a, b);
15. var b;

=>

at 1st line a initialized **a** value 5 and b initialized value 10, after that function foo value of **a** is override with previous value of **a** and not it becomes 15.

At line no. 6 **b** is initialized as 20 but it may not override with previous value because it created with var keyword and with this it will assigned new memory location to **b** which is local variable for function foo.

After that at line no. 8 value of **a** is again override with 25 and **b** value is also override with 30 but at the new location of **b** it may not affected to the global variable value.

That’s it return the value of **a** is 25 & value of **b** is 10.