Module 3

Exercise 02 (Advance JavaScript)

Q 1.)Create a object calculator, which will have methods read,add,subtract and multiply. read method will use prompt to ask two values from user.

A1.)

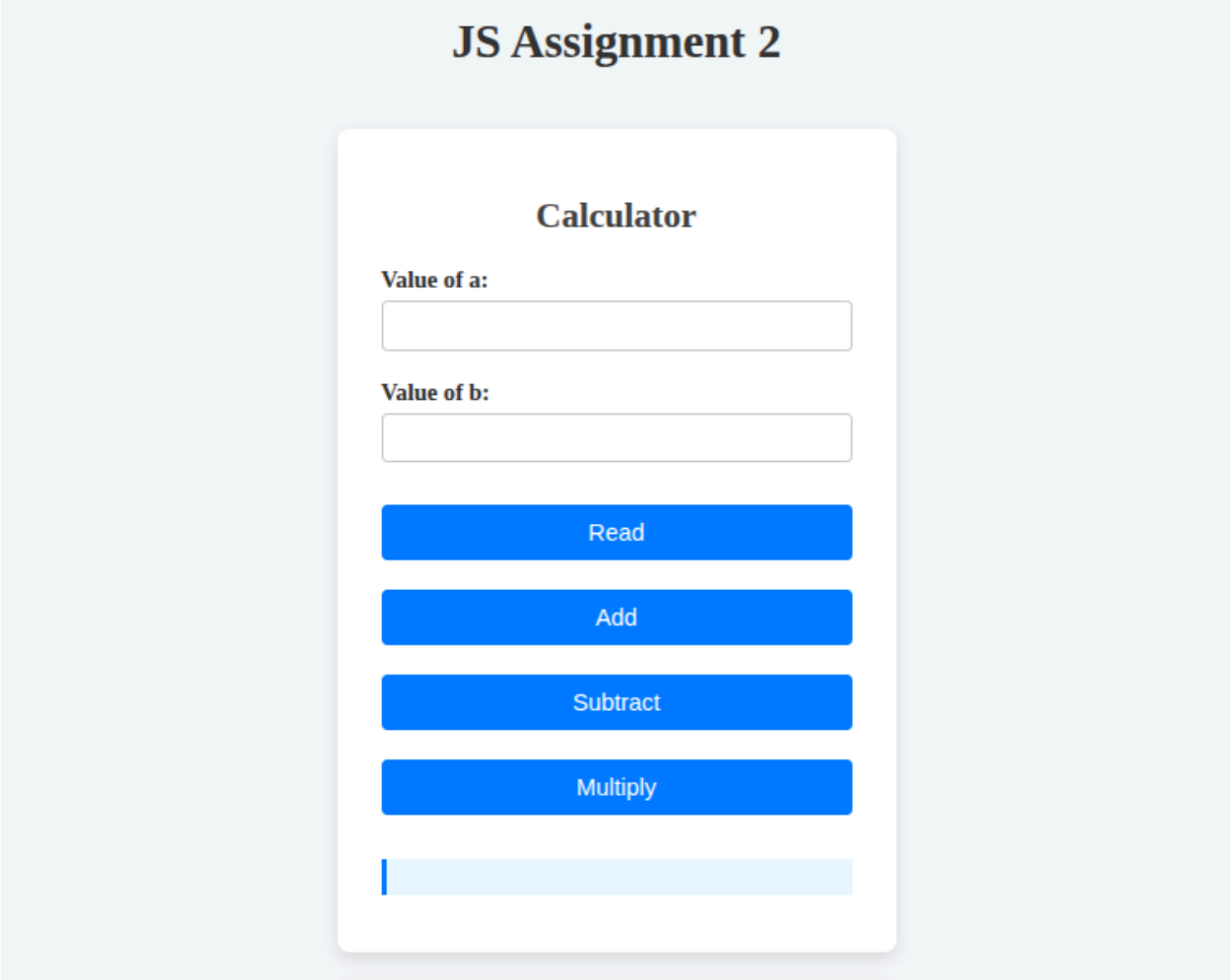
HTML

```html
<h1>JS Assignment 2</h1>

<div class="question">
    <h2>Calculator</h2>

    <form>

        <label>
            Value of a:
            <input type="number" id="first-number" readonly>
        </label><br><br>

        <label>
            Value of b:
            <input type="number" id="second-number" readonly>
        </label><br><br>

        <button type="button" onclick="calculator.read()" id="read-btn">Read</button>

        <button type="button" id="add-btn" onclick="calculator.add()">Add</button>
        <button id="subtract-btn" type="button" onclick="calculator.sub()">Subtract</button>
        <button id="mutiply-btn" type="button" onclick="calculator.multiply()">Multiply</button>
    </form>

    <p class="result" id="result"></p>
</div>
```

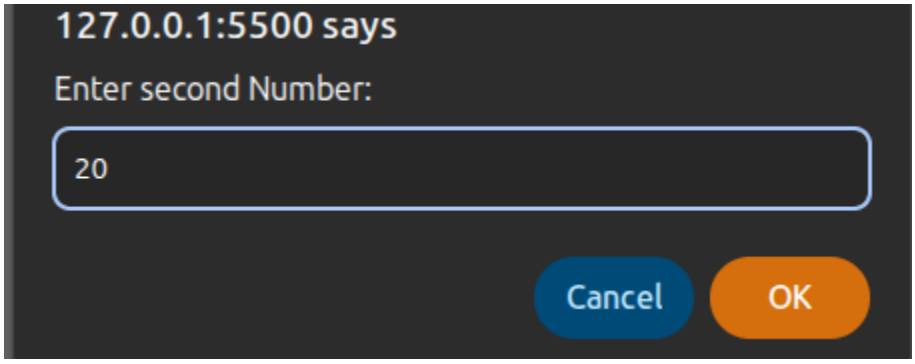
JS

```
var calculator={
    num1:0,
    num2:0,
    read:function(){
        this.num1=Number(prompt("Enter first Number:"));
        this.num2=Number(prompt("Enter second Number:"));
        document.getElementById("first-number").value=this.num1;
        document.getElementById("second-number").value=this.num2
    },
    add:function(){
        document.getElementById("result").innerHTML=`The sum of ${this.num1} & ${this.num2} is ${this

    },
    sub:function(){
        document.getElementById("result").innerHTML=`The difference of ${this.num1} & ${this.num2} is

    },
    multiply:function(){
        document.getElementById("result").innerHTML=`The multiplication of ${this.num1} & ${this.num2

    }
}
```

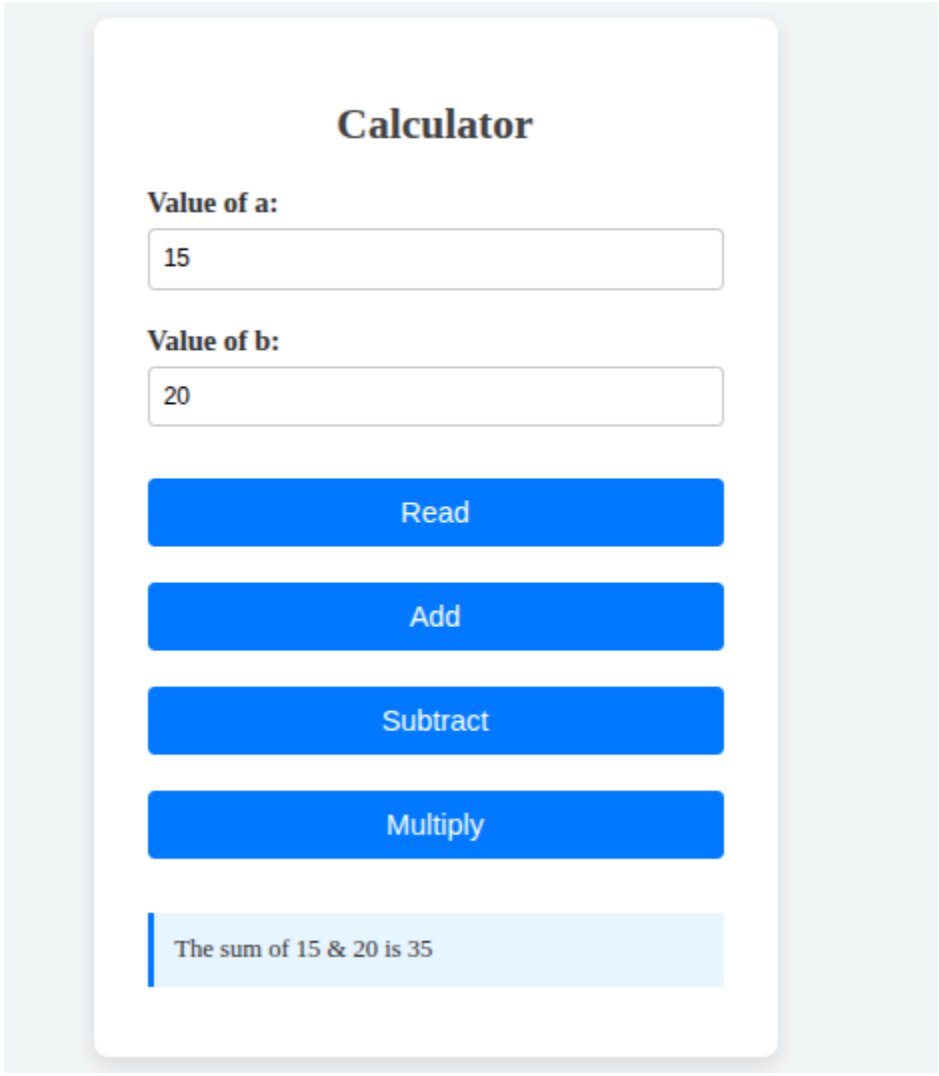
Output



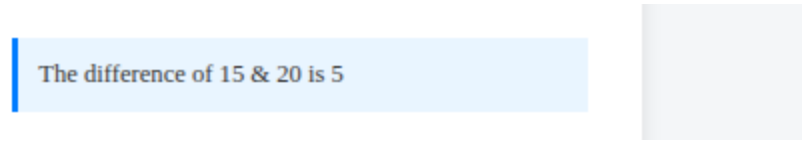
Read


127.0.0.1:5500 says
Enter second Number:
20
Cancel
OK

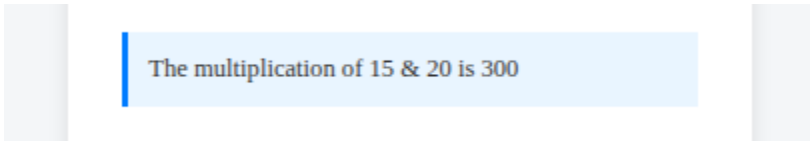
Add


Calculator
Value of a:
15
Value of b:
20
Read
Add
Subtract
Multiply
The sum of 15 & 20 is 35

Sub



Multiply



Q 2.)Create an object temperatureConverter with methods:

- read() → Asks the user to enter a temperature in Celsius.
- toFahrenheit() → Converts it to Fahrenheit.
- toKelvin() → Converts it to Kelvin.
- display() → Displays the results in the console.

A2.)

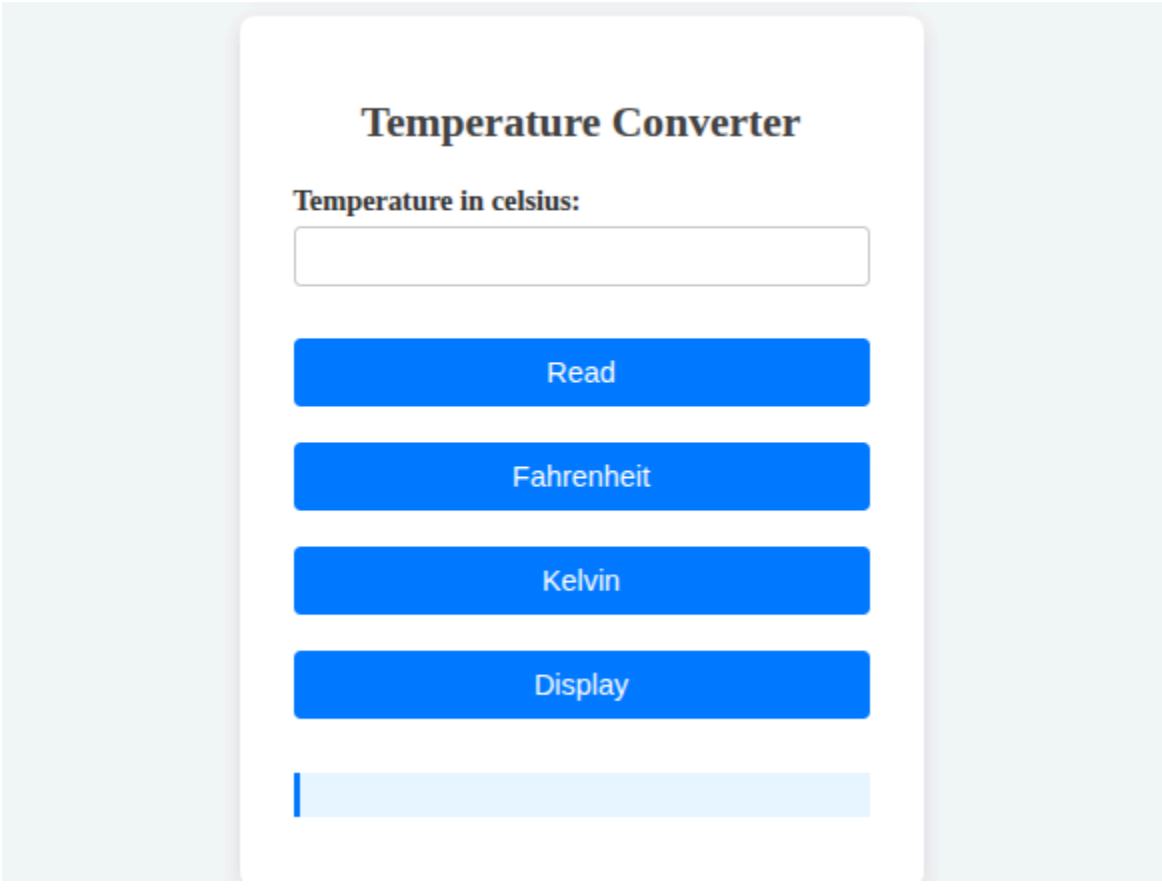
Html

```
<div class="question">
    <h2>Temperature Converter</h2>

    <form>

        <label>
            Temperature in celsius:
            <input type="number" id="temperature-celsius" readonly>
        </label><br><br>

        <button type="button" onclick="temperatureConverter.read()" id="read-temperature">Read</butt

        <button type="button" id="fahrenheit" onclick="temperatureConverter.toFahrenheit()">Fahrenhe
        <button id="kelvin" type="button" onclick="temperatureConverter.toKelvin()">Kelvin</button>
        <button id="display" type="button" onclick="temperatureConverter.display()">Display</button>
    </form>

    <p class="result" id="temperature-result"></p>
</div>
```

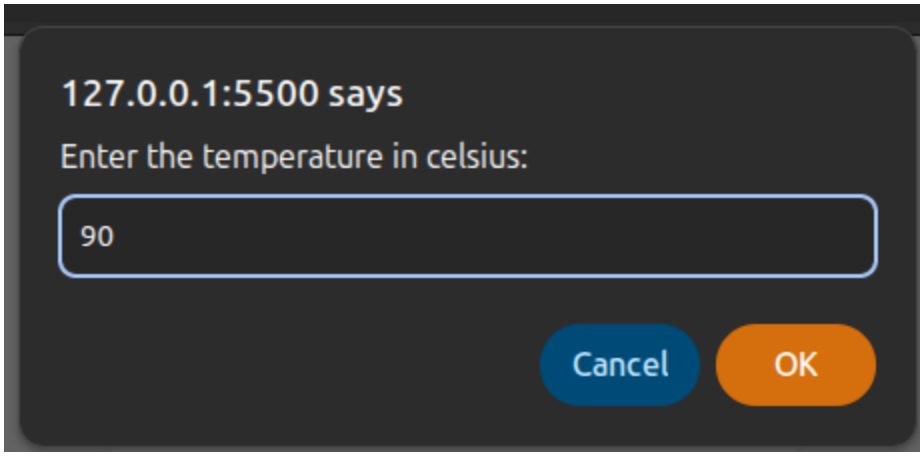
Js

```
var temperatureConverter={
    temperature:null,
    fahrenheit:null,
    kelvin:null,
    read:function(){
        this.temperature=Number(prompt("Enter the temperature in celsius:"));
        document.getElementById("temperature-celsius").value=this.temperature;
    },
    toFahrenheit:function(){
        this.fahrenheit=(this.temperature*9)/5 + 32;
        document.getElementById("temperature-result").innerHTML=`The Fahrenheit of ${this.temperature
    },
    toKelvin:function(){
        this.kelvin=this.temperature+273.15;
        document.getElementById("temperature-result").innerHTML=`The Kelvin of ${this.temperature}°C

    },
    display:function(){
        document.getElementById("temperature-result").innerHTML=`The Temperature is ${this.temperatur
    }
}
```

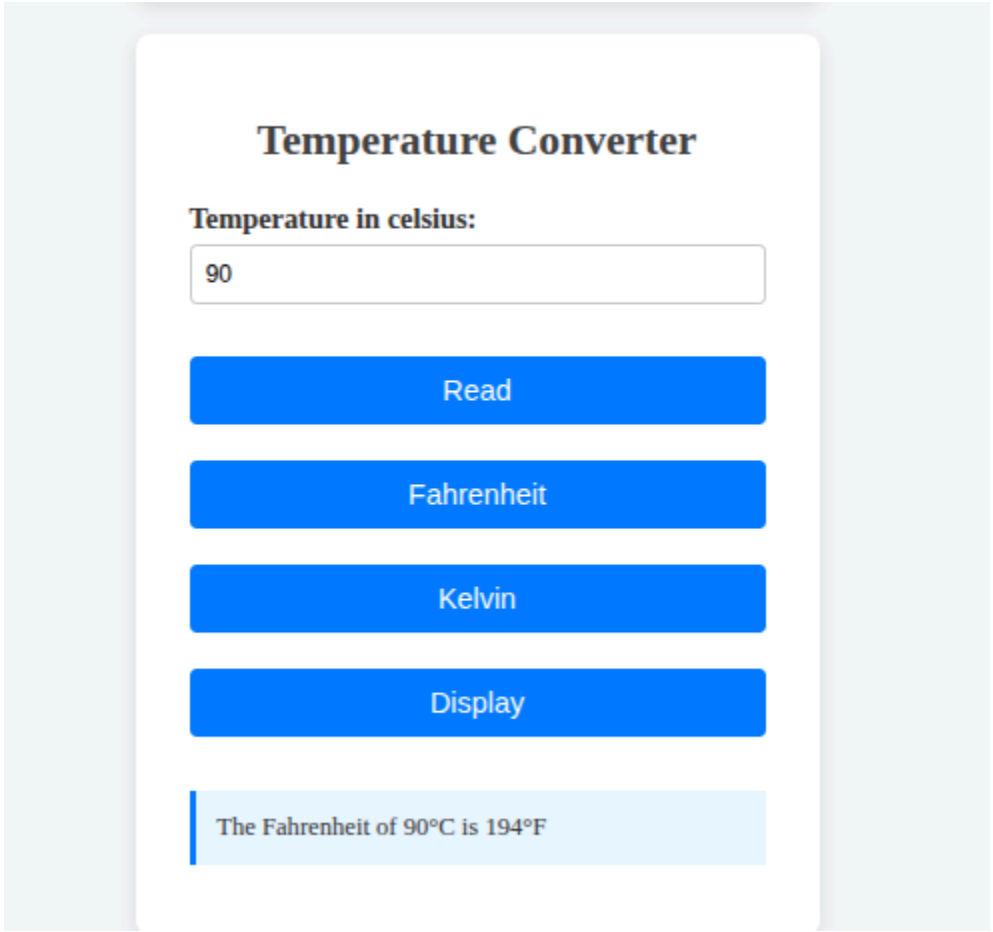
Output



Read



Fahrenheit



Kelvin

The Kelvin of 90°C is 363.15K

Display

The Temperature is 90°C , 194°F and 363.15K

Q3.) Tasks:

```
var x = 5;

function first() {
  console.log(x); // Line 1
  var y = 10;

  function second() {
    console.log(y); // Line 2
    console.log(z); // Line 3
    var z = 20;
  }

  second();
}

first();
console.log(y); // Line 4
```

- Predict the output of the given snippet.

  A.) console.log(x) gives 5//Line1

  console.log(y) gives 10//Line2

  console.log(z) gives undefined//Line3

  console.log(y) gives a reference error//Line4

- Explain how hoisting affects the execution of console.log(z) in second().

  A.) In second function at execution time the declaration of var z=20; will get sifted to the top of the stack and the initialization

will remain at the same line making the value of the var z undefined. This whole process is called Hoisting and console.log(z) gives undefined

```
function second(){
        var z;
        console.log(y);
        console.log(z);
        z=20;
}
```

- Explain the scope chain for console.log(y) in second().

  A.) For the console.log(y) scope chaining is applied scope chaining is a feature of javascript which let a function access a variable present inside its parent or grandparent or great grandparent i.e It is basically first search for variable insides own scope if it did not find it will move to parent scope this process will continue till the variable has found or the scope ends giving a Reference Error.

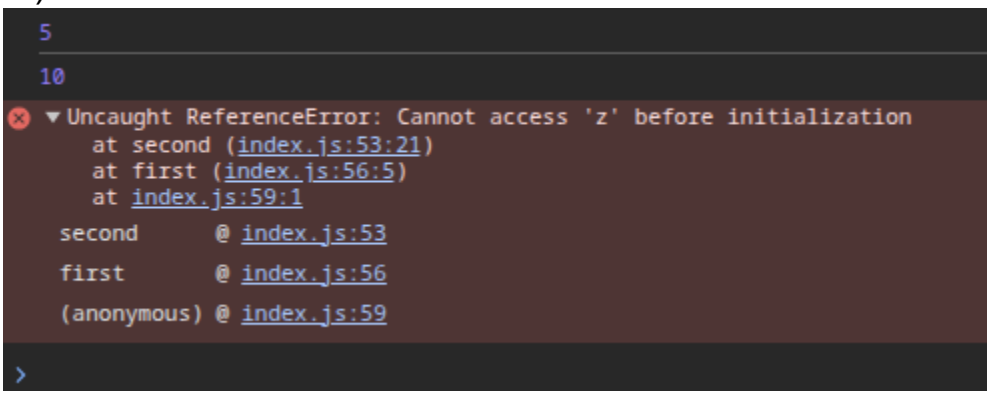
  For console.log(y) it will search in second() function first then move to the first() function and find it there giving a value of 10

- What happens when console.log(y) is executed outside first() (Line 4)? Why?

  A.) When the console.log(y) is executed outside first() we will get a Reference Error that is because we are trying to access the var y outside its scope(The var are function scope the is there are only available till the execution of the function or block of code but can be accessed by the child due to scope chaining). So Due to function scoping of var we will get the reference error on Line 4

- Modify the code to use let instead of var and observe any differences.

A.)



This is the output if we will change the var with let
Getting a output as follows
console.log(x) as 5 //Line1
console.log(y) as 10 //Line2
console.log(z) as Reference Error //Line3
console.log(y) will not get executed due to error //Line4

We can clearly see that the let should be initialized before accessing it otherwise we will get Reference Error . It is due the temporal dead zone it is a state where the let and const are present without initialization in this case the hoisting of let z make it temporal dead zone.