1. Functions and Prototyping.

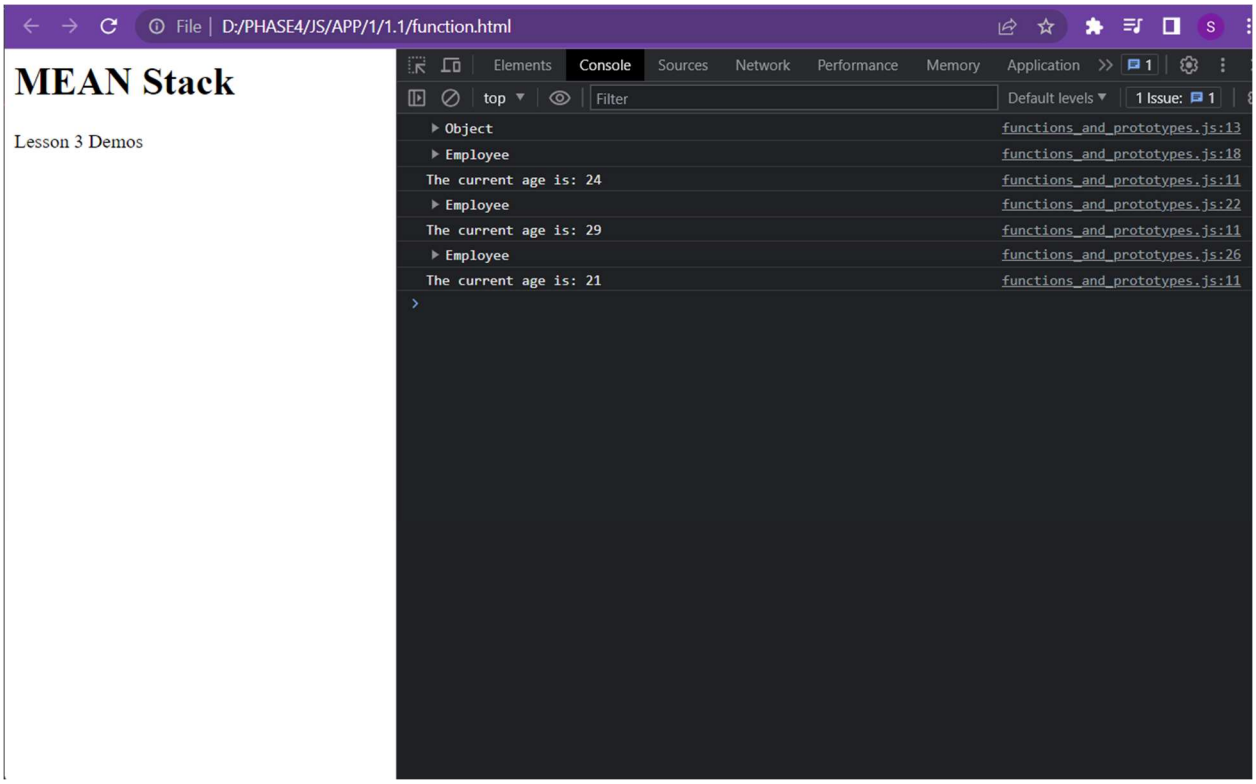
Function.html

```
<html>
    <body>
        <h1>MEAN Stack</h1>
        <p> Lesson 3 Demos </p>

        <script src="functions_and_prototypes.js"></script>


    </body>
</html>
```

functions_and_prototypes.js

```
// function constructor
function Employee(name, designation, yearOfBirth){
    this.name= name;
    this.designation= designation;
    this.yearOfBirth= yearOfBirth;
}

// creating calculateAge() method to the Prototype property
Employee.prototype.calculateAge= function(){
    console.log('The current age is: '+(2019- this.yearOfBirth));
}
console.log(Employee.prototype);


// creating Objects
let Emp1= new Employee('Alex', 'Junior Tester', 1995);
console.log(Emp1) ;
Emp1.calculateAge();

let Emp2= new Employee('Dexter', 'Senior Software Developer', 1990);
console.log(Emp2)
Emp2.calculateAge();

let Emp3= new Employee('Annie', 'Junior HR', 1998);
console.log(Emp3)
Emp3.calculateAge();
```

OUTPUT


File | D:/PHASE4/JS/APP/1/1.1/function.html
MEAN Stack
Lesson 3 Demos
Elements
Console
Sources
Network
Performance
Memory
Application
top
Filter
Default levels
1 Issue: 1
Object
functions_and_prototypes.js:13
Employee
functions_and_prototypes.js:18
The current age is: 24
functions_and_prototypes.js:11
Employee
functions_and_prototypes.js:22
The current age is: 29
functions_and_prototypes.js:11
Employee
functions_and_prototypes.js:26
The current age is: 21
functions_and_prototypes.js:11

2. Working with Functions.

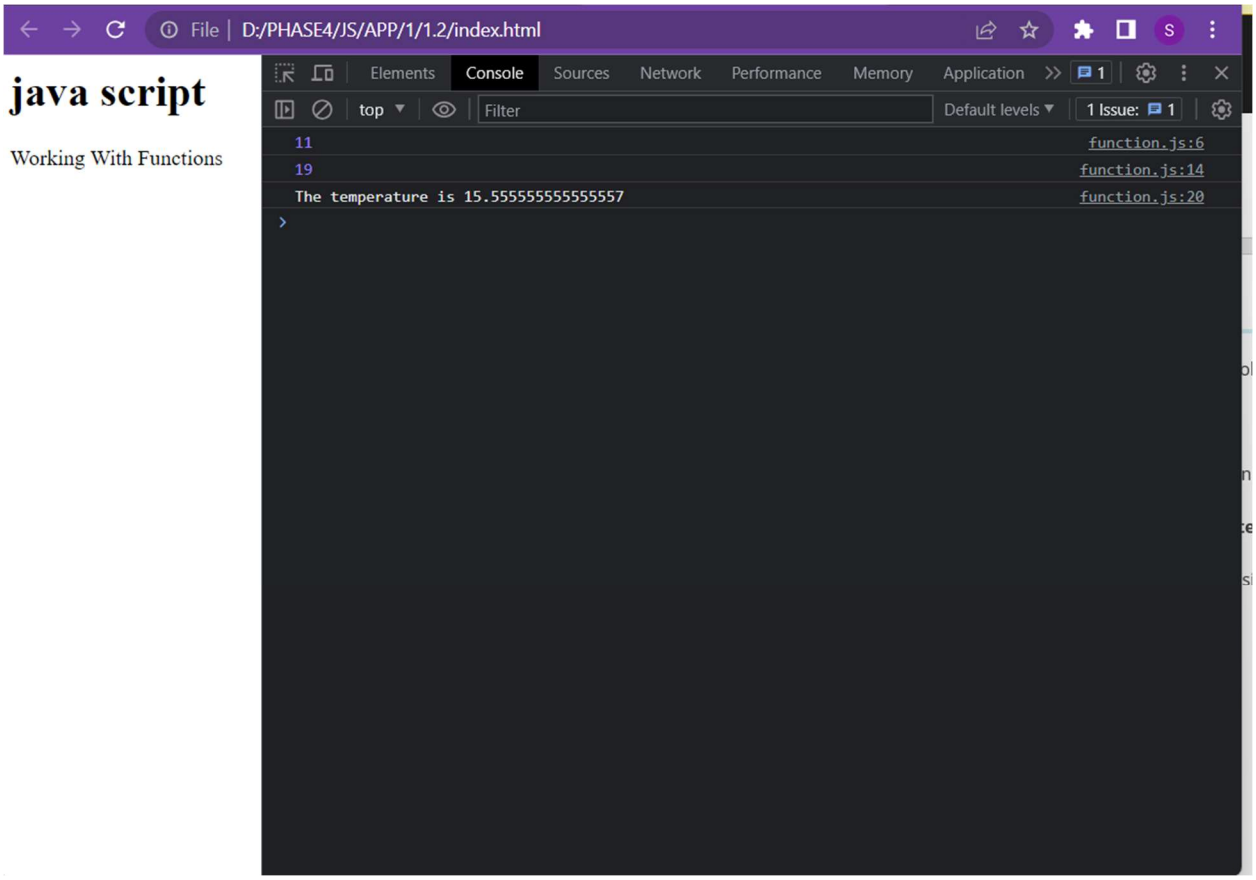
Index.html

```
<html>
    <body>
        <h1>java script</h1>
        <p> Working With Functions </p>

        <script src="function.js"></script>


    </body>
</html>
```

Function.js

```
    var x = (2 * 3) + 5;
    var y = 3 * 4;

    var result = myFunction(2,3);
    console.log(result);

    function myFunction(num1, num2) {
        var a = num1 * num2;
        var b = num1 + num2;
        return(a + b);
    }

    console.log( myFunction(3, 4));

    function toCelcius(f){
            return (5/9) * (f-32);
    }

    console.log("The temperature is "+ toCelcius(60));
```

Output


File | D:/PHASE4/JS/APP/1/1.2/index.html
java script
Working With Functions
Elements
Console
Sources
Network
Performance
Memory
Application
top
Filter
Default levels
1 Issue: 1
11
function.js:6
19
function.js:14
The temperature is 15.555555555555557
function.js:20

3. IIFEs, Callbacks, and Closures.

Index.html

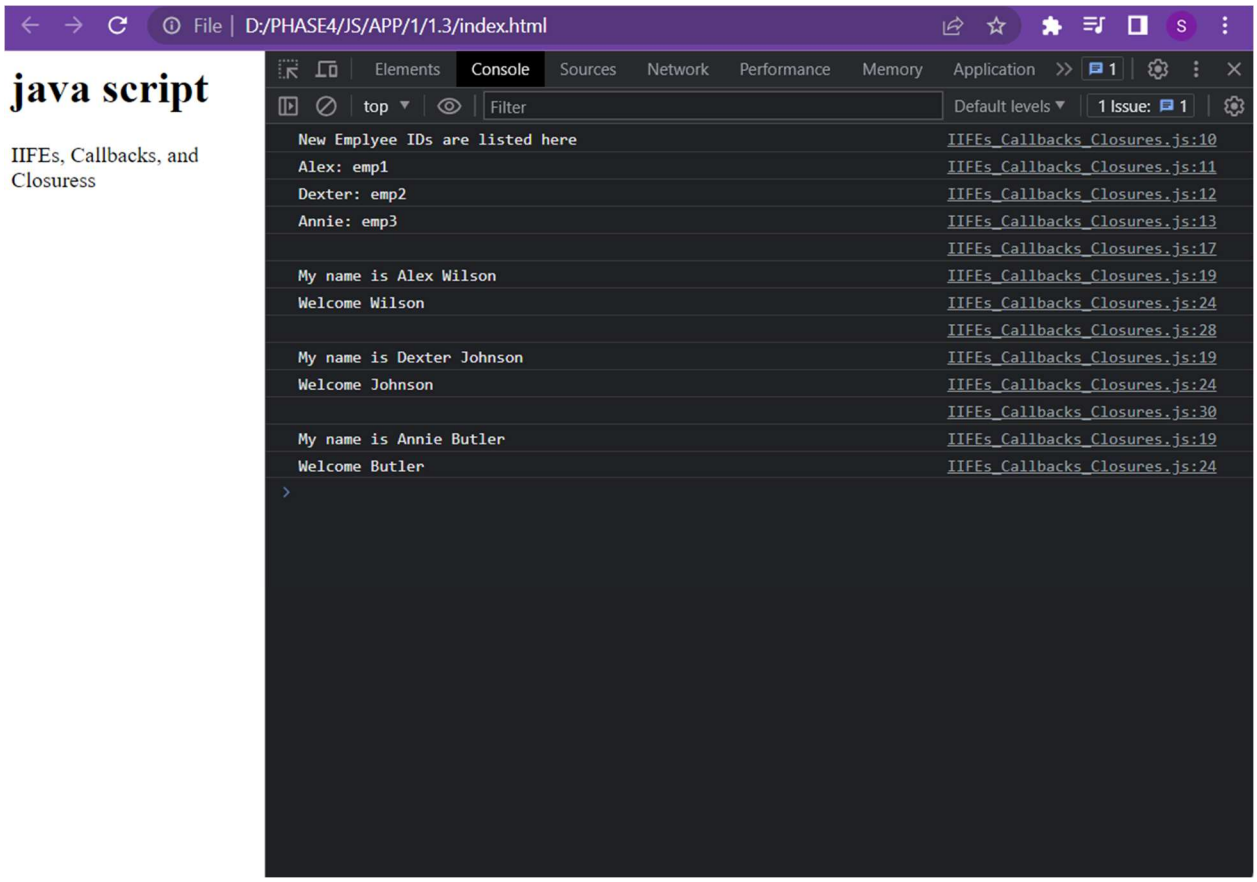
```
<html>
    <body>
        <h1>java script</h1>
        <p> IIFEs, Callbacks, and Closuress </p>

        <script src="IIFEs_Callbacks_Closures.js"></script>


    </body>
</html>
```

IIFEs_Callbacks_Closures.js

```
//IIFE and Closure
const empId = (function() {
    let count = 0;
    return function() {
      ++count;
      return `emp${count}`;
    };
  })();

  console.log("New Emplyee IDs are listed here");
  console.log("Alex: "+empId());
  console.log("Dexter: "+empId());
  console.log("Annie: "+empId());


  //Callbacks
  console.log("\n"); //to start the output from the neext line
  function fullName(firstName, lastName, callback){
    console.log("My name is " + firstName + " " + lastName);
    callback(lastName);
  }

  var greeting = function(ln){
    console.log('Welcome ' + ln);
  };
```

```
fullName("Alex", "Wilson", greeting);
console.log("\n");
fullName("Dexter", "Johnson", greeting);
console.log("\n");
fullName("Annie", "Butler", greeting);
```

OUTPUT



4. Maps and Classes.

Index.html

```
<html>
    <body>
        <h1>Java Script</h1>
        <p>Maps and Classes. </p>

        <script src="maps_and_classes.js"></script>


    </body>
</html>
```

maps_and_classes.js

```
var map1 = new Map();
map1.set("first name", "Robb");
map1.set("last name", "Stark");
map1.set("friend 1","Bran")
    .set("friend 2","Arya");
console.log(map1);
console.log("map1 has friend 3 ? " + map1.has("friend 3"));
console.log("get value for key = friend 3 - "+ map1.get("friend 3"));
console.log("delete element with key = friend 2 - " + map1.delete("friend 2"));
map1.clear();
console.log(map1);
class Employee
{
    constructor(id,name)
    {
      this.id=id;
      this.name=name;
    }
    detail()
    {
  document.writeln(this.id+" "+this.name+"<br>")
    }
  }
//passing object to a variable
var e1=new Employee(101,"Michael");
```

```
var e2=new Employee(102,"Bob");
e1.detail();
e2.detail();
```

OUTPUT