



IFT544: Middleware Prog & Database Security
Module 2 Assignment: JavaScript language

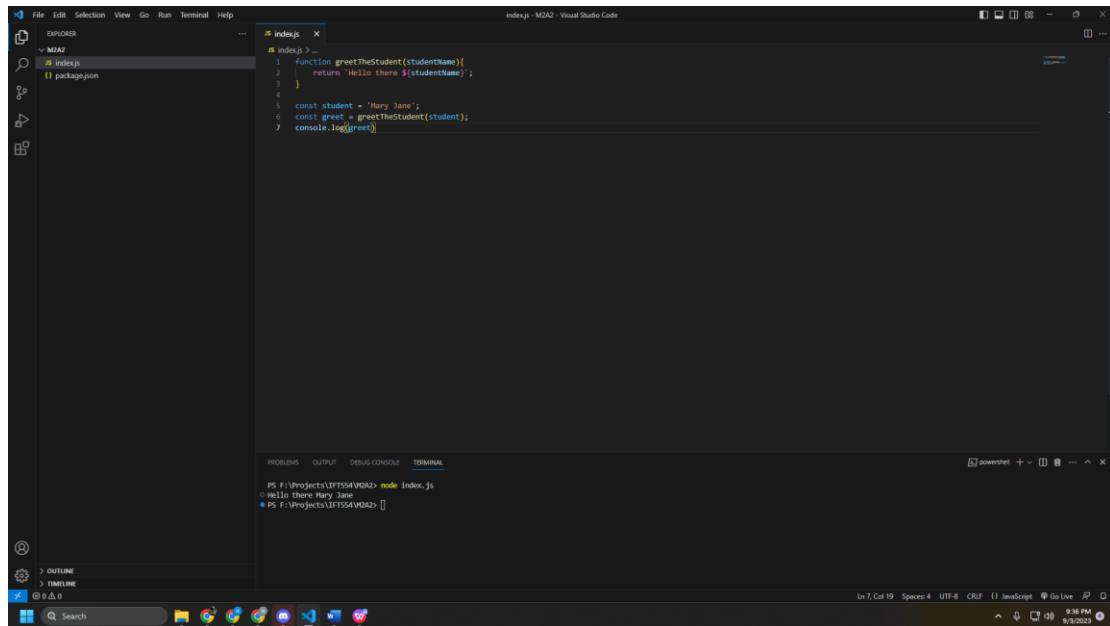
Author: Nived Abdulsathar (1225125746)

Instructor: **Dinesh Sthapit**

Date of Submission: September 3, 2023

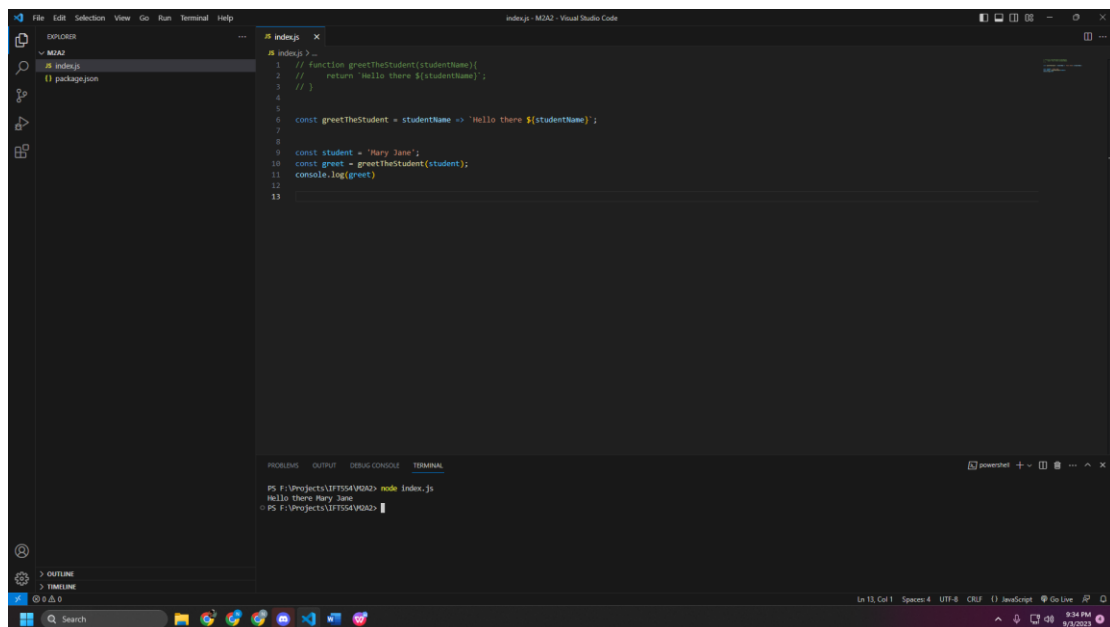
Introduction to JavaScript -1

```
function greetTheStudent(studentName){  
    return `Hello there ${studentName}`;  
}  
  
const student = 'Mary Jane';  
const greet = greetTheStudent(student);  
console.log(greet)
```



JavaScript Function Refactoring 2

```
// function greetTheStudent(studentName){  
//     return `Hello there ${studentName}`;  
// }  
  
const greetTheStudent = studentName => `Hello there ${studentName}`;  
  
const student = 'Mary Jane';  
const greet = greetTheStudent(student);  
console.log(greet)
```



The screenshot shows the Visual Studio Code editor with a file named `index.js` open. The code in the editor is as follows:

```
1 // function greetTheStudent(studentName){  
2 //     return `Hello there ${studentName}`;  
3 // }  
4  
5  
6 const greetTheStudent = studentName => `Hello there ${studentName}`;  
7  
8  
9 const student = 'Mary Jane';  
10 const greet = greetTheStudent(student);  
11 console.log(greet)  
12  
13
```

The Explorer sidebar on the left shows the project structure with `index.js` and `package.json` files. The Terminal at the bottom shows the command `node index.js` being executed, resulting in the output `Hello there Mary Jane`.

JavaScript Refactoring 3

```
// function greetTheStudent(studentName){
//     return `Hello there ${studentName}`;
// }

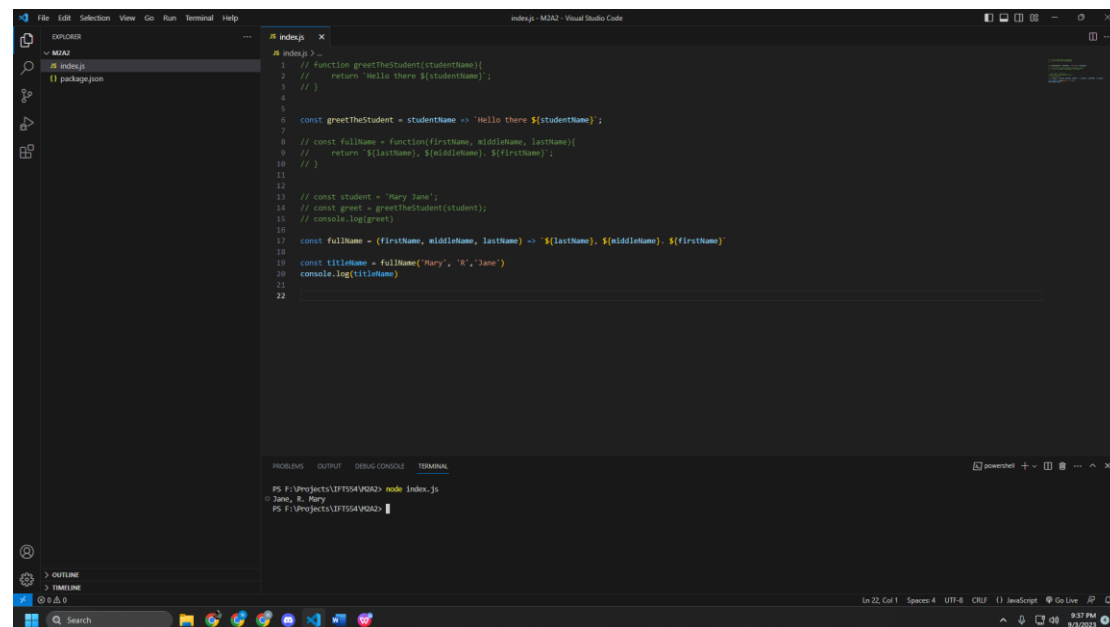
const greetTheStudent = studentName => `Hello there ${studentName}`;

// const fullName = function(firstName, middleName, lastName){
//     return `${lastName}, ${middleName}. ${firstName}`;
// }

// const student = 'Mary Jane';
// const greet = greetTheStudent(student);
// console.log(greet)

const fullName = (firstName, middleName, lastName) => `${lastName},
${middleName}. ${firstName}`

const titleName = fullName('Mary', 'R','Jane')
console.log(titleName)
```



JavaScript Variables 4

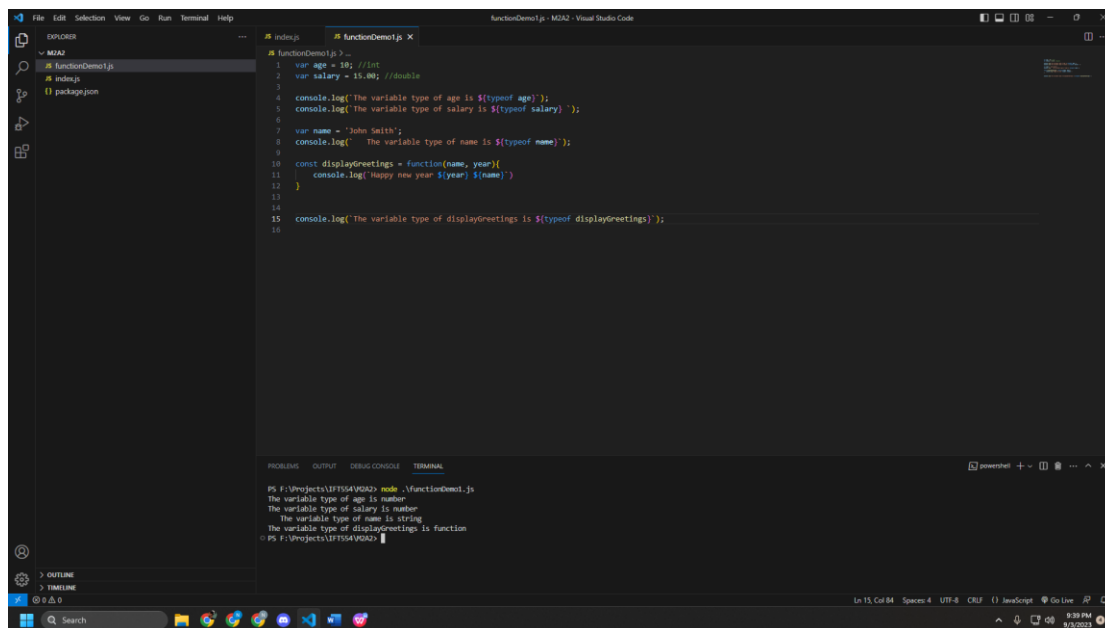
```
var age = 10; //int
var salary = 15.00; //double

console.log(`The variable type of age is ${typeof age}`);
console.log(`The variable type of salary is ${typeof salary} `);

var name = 'John Smith';
console.log(`    The variable type of name is ${typeof name}`);

const displayGreetings = function(name, year){
    console.log(`Happy new year ${year} ${name}`)
}

console.log(`The variable type of displayGreetings is ${typeof
displayGreetings}`);
```



JavaScript Objects 5

```
var age = 10; //int
var salary = 15.00; //double

// console.log(`The variable type of age is ${typeof age}`);
// console.log(`The variable type of salary is ${typeof salary} `);

// var name = 'John Smith';
// console.log(`    The variable type of name is ${typeof name}`);

const displayGreetings = function(name, year){
    console.log(`Happy new year ${year} ${name}`)
}

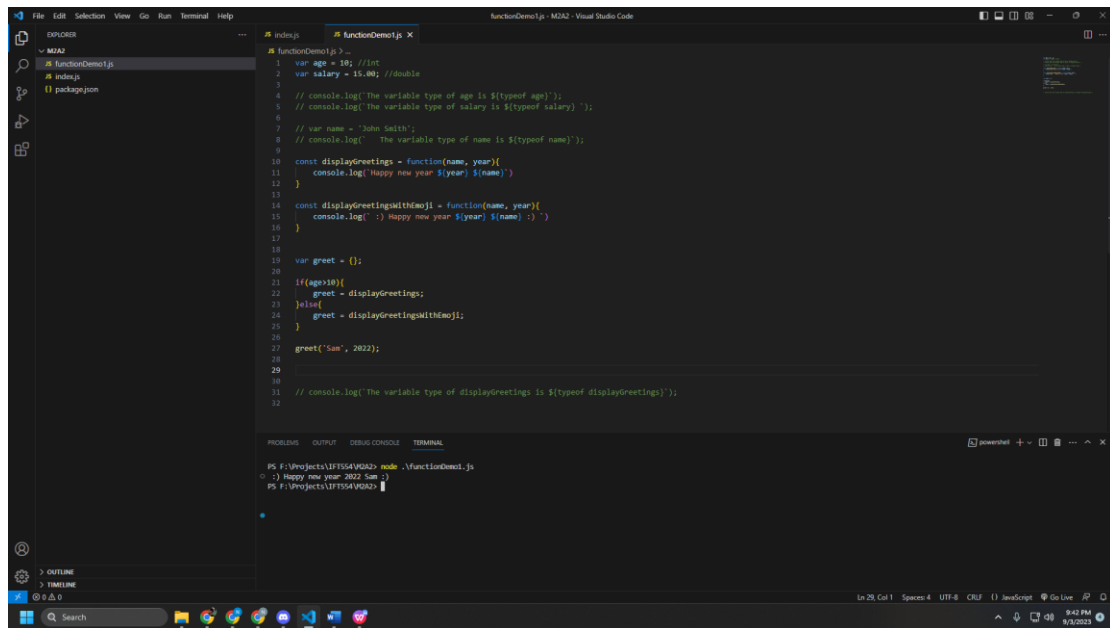
const displayGreetingsWithEmoji = function(name, year){
    console.log(` :) Happy new year ${year} ${name} :) `)
}

var greet = {};

if(age>10){
    greet = displayGreetings;
}else{
    greet = displayGreetingsWithEmoji;
}

greet('Sam', 2022);

// console.log(`The variable type of displayGreetings is ${typeof
displayGreetings}`);
```



JavaScript Parameter (Function as a parameter) 6

```
var age = 10; //int
var salary = 15.00; //double

// console.log(`The variable type of age is ${typeof age}`);
// console.log(`The variable type of salary is ${typeof salary} `);

// var name = 'John Smith';
// console.log(`    The variable type of name is ${typeof name}`);

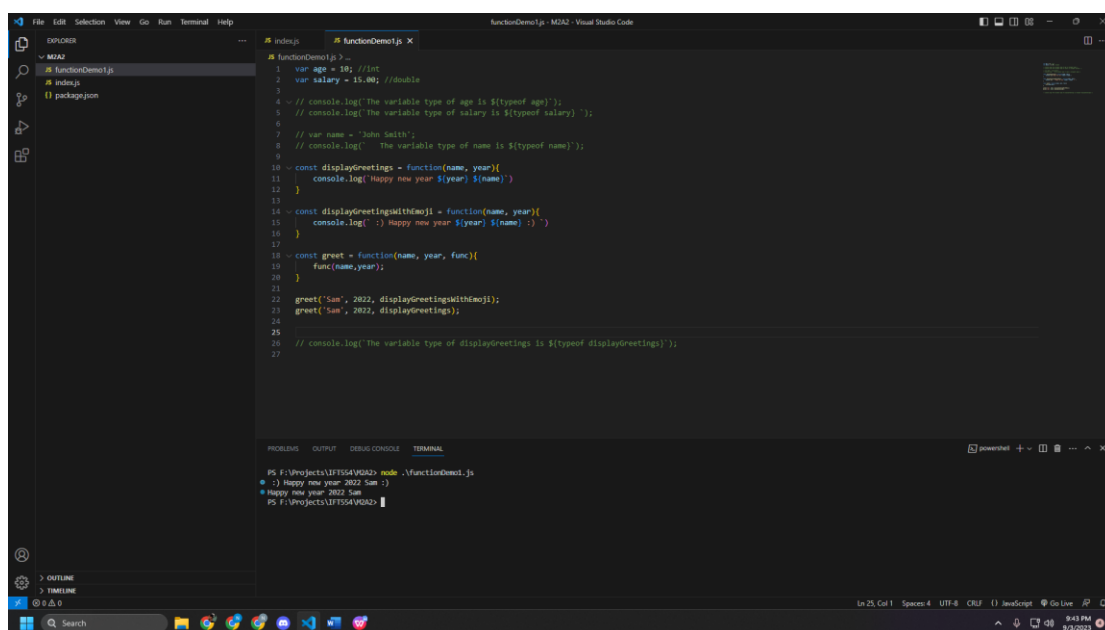
const displayGreetings = function(name, year){
    console.log(`Happy new year ${year} ${name}`)
}

const displayGreetingsWithEmoji = function(name, year){
    console.log(` :) Happy new year ${year} ${name} :) `)
}

const greet = function(name, year, func){
    func(name,year);
}

greet('Sam', 2022, displayGreetingsWithEmoji);
greet('Sam', 2022, displayGreetings);

// console.log(`The variable type of displayGreetings is ${typeof displayGreetings}`);
```



```
functionDemo1.js
1 var age = 10; //int
2 var salary = 15.00; //double
3
4 // console.log(`The variable type of age is ${typeof age}`);
5 // console.log(`The variable type of salary is ${typeof salary} `);
6
7 // var name = 'John Smith';
8 // console.log(`    The variable type of name is ${typeof name}`);
9
10 const displayGreetings = function(name, year){
11     console.log(`Happy new year ${year} ${name}`)
12 }
13
14 const displayGreetingsWithEmoji = function(name, year){
15     console.log(` :) Happy new year ${year} ${name} :) `)
16 }
17
18 const greet = function(name, year, func){
19     func(name,year);
20 }
21
22 greet('Sam', 2022, displayGreetingsWithEmoji);
23 greet('Sam', 2022, displayGreetings);
24
25 // console.log(`The variable type of displayGreetings is ${typeof displayGreetings}`);
26
27
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS F:\Projects\JFTSSA\VS2022> node .\functionDemo1.js
Happy new year 2022 Sam :)
Happy new year 2022 Sam
PS F:\Projects\JFTSSA\VS2022>
```


JavaScript Master Document Reference (required) 7

The screenshot shows the MDN web docs page for 'Looping code'. The page is in a dark theme. The left sidebar contains a navigation menu with categories like 'Functions', 'Web forms', 'Accessibility', and 'Performance'. The main content area has a heading 'Looping code' and a subheading 'Here's the JavaScript code that implements this example:'. Below this is a code block with JavaScript code for a canvas drawing application. The right sidebar has a section 'In this article' with links to various topics related to looping.

Canvas (MS) | Course Modules | Module 2 Assignment: JavaScript | Module 2 Lab 1: JavaScript | No.23A/MOAT | Looping code - Learn web dev: |

developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Looping_code

mdn web docs | References | Guides | Plus | Blog | Play | AI Help | Theme | Log in | Sign up for free

Guides > JavaScript — Dynamic client-side scripting > JavaScript building blocks > Looping code | English (US)

Filter

- Looping code
- Functions — reusable blocks of code
- Build your own function
- Function return values
- Introduction to events
- Image gallery
- Introducing JavaScript objects
- Asynchronous JavaScript
- Client-side web APIs
- Web forms — Working with user data
- Core forms learning pathway
- Advanced forms articles
- Accessibility — Make the web usable by everyone
- Accessibility guides
- Performance — Making websites fast and responsive
- Performance guides

Here's the JavaScript code that implements this example:

```
JS
const btn = document.querySelector("button");
const canvas = document.querySelector("canvas");
const ctx = canvas.getContext("2d");

document.addEventListener("DOMContentLoaded", () => {
  canvas.width = document.documentElement.clientWidth;
  canvas.height = document.documentElement.clientHeight;
});

function random(number) {
  return Math.floor(Math.random() * number);
}

function draw() {
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  for (let i = 0; i < 100; i++) {
    ctx.beginPath();
    ctx.fillStyle = `rgb(${255 * Math.random()})`;
    ctx.arc(
      random(canvas.width),
```

Play |

In this article

- Why are loops useful?
- Looping through a collection
- The standard for loop
- Exiting loops with break
- Skipping iterations with continue
- while and do...while
- Active learning: Launch countdown
- Active learning: Filling in a guest list
- Which loop type should you use?
- Test your skills!
- Conclusion
- See also

Build apps

Build apps and

9:57 PM 8/3/2024

JavaScript Objects II 8

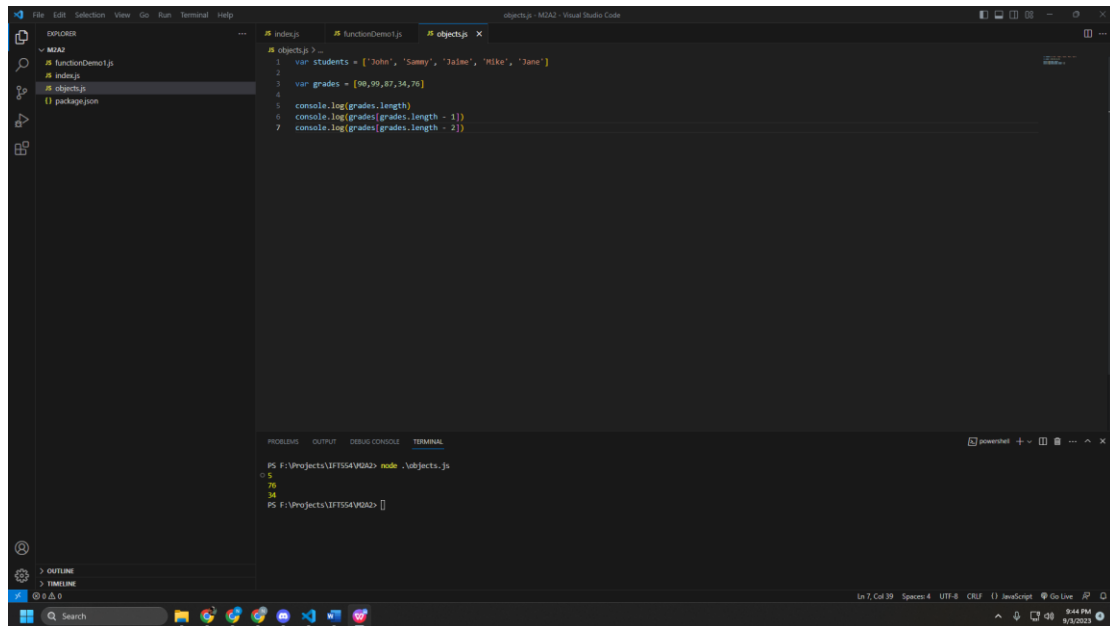
```
var students = ['John', 'Sammy', 'Jaime', 'Mike', 'Jane']
```

```
var grades = [90,99,87,34,76]
```

```
console.log(grades.length)
```

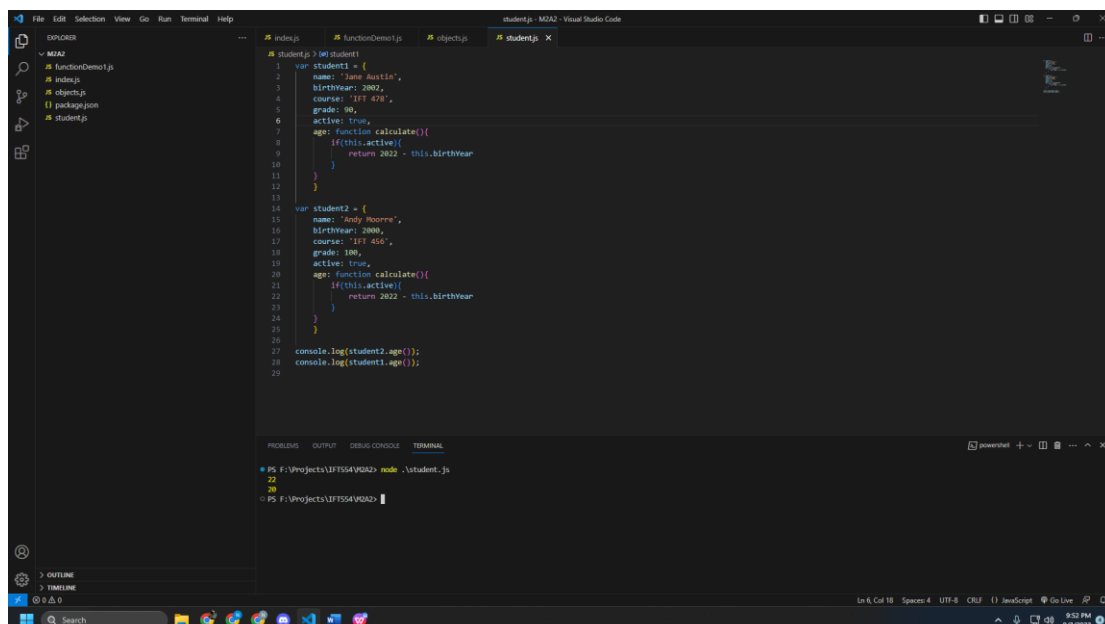
```
console.log(grades[grades.length - 1])
```

```
console.log(grades[grades.length - 2])
```



JavaScripts Arrays 9

```
var student1 = {  
  name: 'Jane Austin',  
  birthYear: 2002,  
  course: 'IFT 478',  
  grade: 90,  
  active: true,  
  age: function calculate(){  
    if(this.active){  
      return 2022 - this.birthYear  
    }  
  }  
}  
  
var student2 = {  
  name: 'Andy Moorre',  
  birthYear: 2000,  
  course: 'IFT 456',  
  grade: 100,  
  active: true,  
  age: function calculate(){  
    if(this.active){  
      return 2022 - this.birthYear  
    }  
  }  
}  
  
console.log(student2.age());  
console.log(student1.age());
```



```

var student1 = {
  name: 'Jane Austin',
  birthYear: 2002,
  course: 'IFT 478',
  grade: 90,
  active: true,
  age: function calculate(){
    if(this.active){
      return 2022 - this.birthYear
    }
  }
}

var student2 = {
  name: 'Andy Moorre',
  birthYear: 2000,
  course: 'IFT 456',
  grade: 100,
  active: false,
  age: function calculate(){
    if(this.active){
      return 2022 - this.birthYear
    }
    else{
      return 0;
    }
  }
}

console.log(student2.age());
console.log(student1.age());

```

```

1  var student1 = {
2    name: 'Jane Austin',
3    birthYear: 2002,
4    course: 'IFT 478',
5    grade: 90,
6    active: true,
7    age: function calculate(){
8      if(this.active){
9        return 2022 - this.birthYear
10     }
11   }
12 }
13
14 var student2 = {
15   name: 'Andy Moorre',
16   birthYear: 2000,
17   course: 'IFT 456',
18   grade: 100,
19   active: false,
20   age: function calculate(){
21     if(this.active){
22       return 2022 - this.birthYear
23     }
24     else{
25       return 0;
26     }
27   }
28 }
29
30 console.log(student2.age());
31 console.log(student1.age());
32

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS F:\Projects\IFT554\VG02> node .\student.js
20
20
PS F:\Projects\IFT554\VG02> node .\student.js
20
20
PS F:\Projects\IFT554\VG02>

```

La 25, Col 22, Spaces 4, UTF-8, CRLF, JavaScript, Go Live, 9:53 PM, 8/5/2022

Accessing Elements of JavaScript Object Arrays 10

```
var students = []
var student1 = {
  name: 'Jane Austin',
  birthYear: 2002,
  course: 'IFT 478',
  grade: 90,
  active: true,
  age: function calculate(){
    if(this.active){
      return 2022 - this.birthYear
    }
  }
}

var student2 = {
  name: 'Andy Moorre',
  birthYear: 2000,
  course: 'IFT 456',
  grade: 100,
  active: false,
  age: function calculate(){
    if(this.active){
      return 2022 - this.birthYear
    }
    else{
      return 0;
    }
  }
}

students.push(student1);
students.push(student2);

students.forEach((item)=> console.log(item.age()));
console.log(students)
```

