Lesson 04 Demo 07

Demonstrating Pass by Reference and Pass by Value Methods

Objective: To demonstrate the use of pass by reference and pass by value methods for manipulating the values of the data structures in Node.js

Tools required: Visual Studio Code and Node Package Manager

Prerequisites: Basic Linux and NPM commands

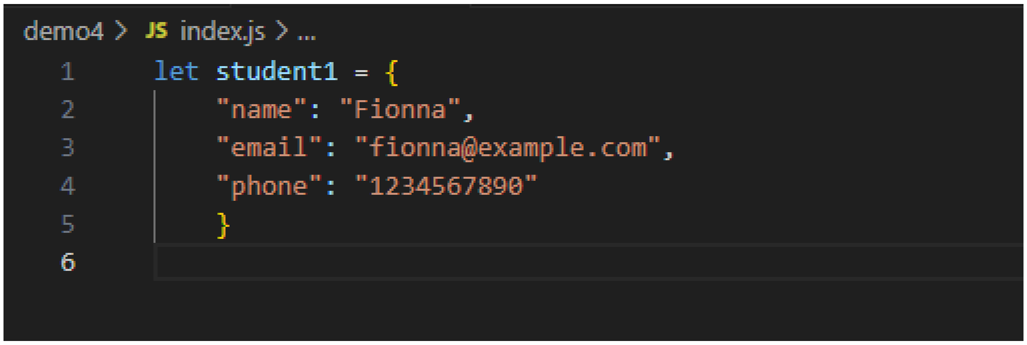
Steps to be followed:

1. Create an object using the pass by reference method
2. Create a variable using the pass by value method

Step 1: Create an object using the pass by reference method

1.1 Navigate inside the project directory and create an object with following details in the **index.js** file:

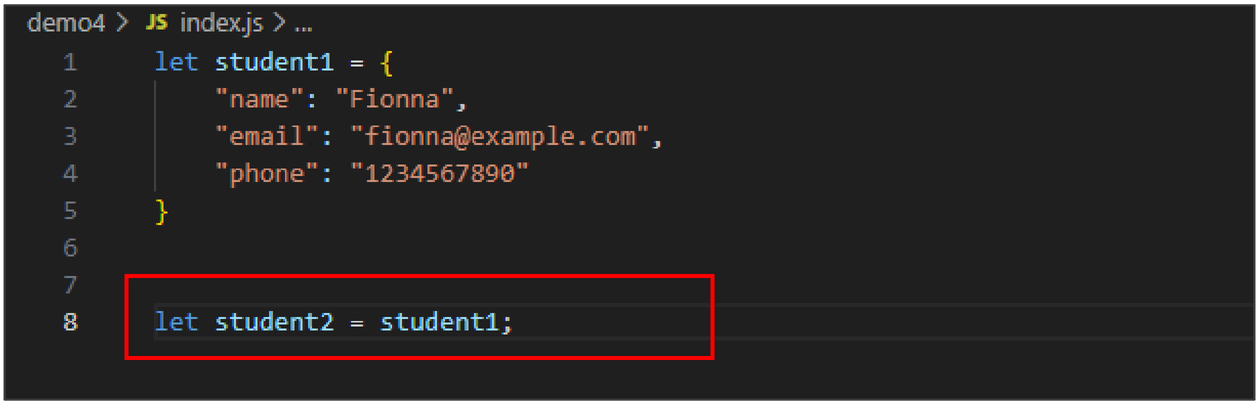
```
let student1 = {
   "name": "Fionna",
   "email": "fionna@example.com",
   "phone": "1234567890"
   }
```



```
let student1 = {
    "name": "Fionna",
    "email": "fionna@example.com",
    "phone": "1234567890"
    }
```

1.2 Copy **the student1** object into the **student2** object, which is a reference copy

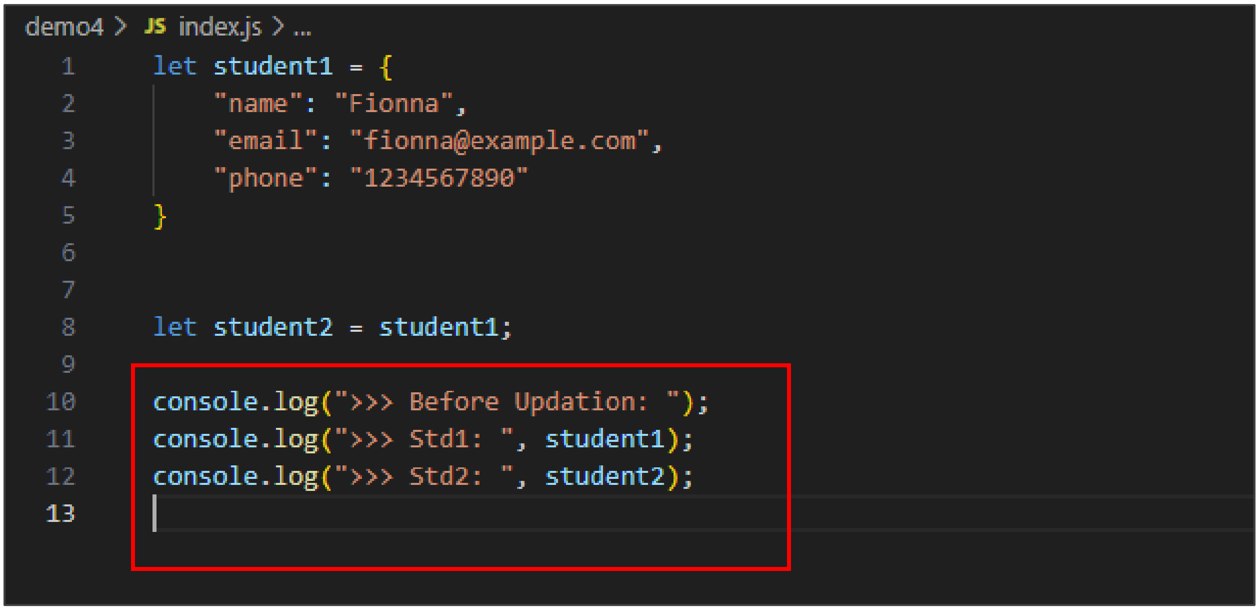
let student2 = student1;



```
let student1 = {
    "name": "Fionna",
    "email": "fionna@example.com",
    "phone": "1234567890"
}


let student2 = student1;
```

1.3 Print the values of both objects before updating the values:

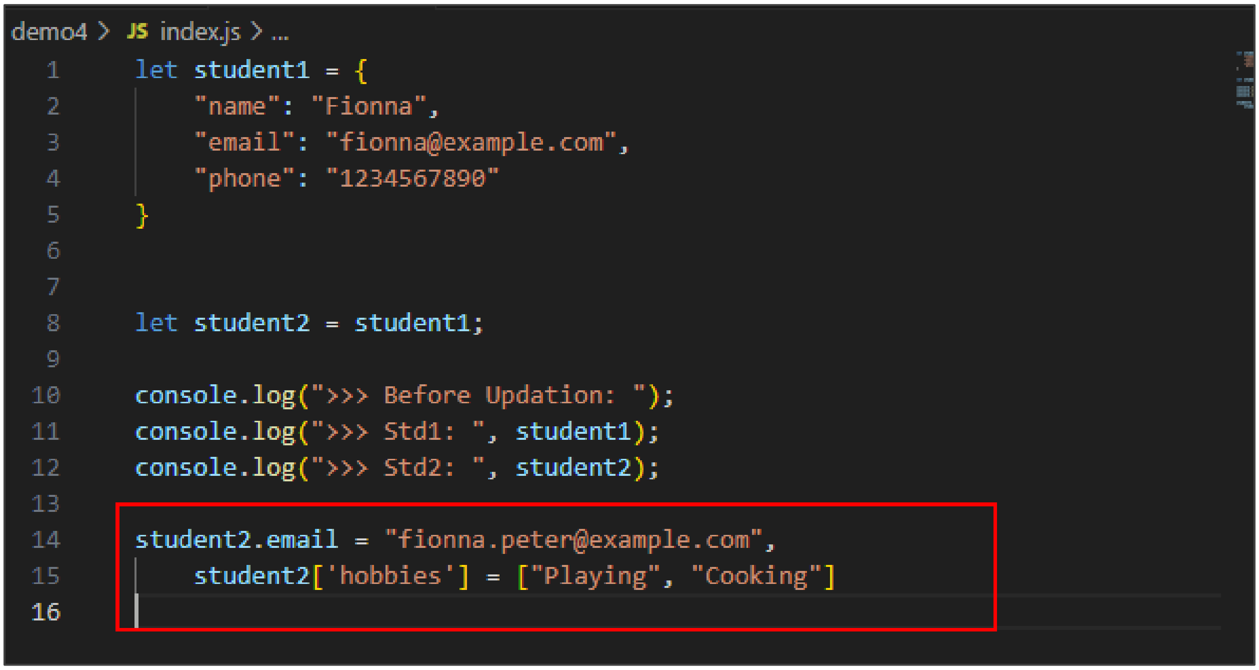
console.log(">>> Before Updation: ");
console.log(">>> Std1: ", student1);
console.log(">>> Std2: ", student2);



```
let student1 = {
    "name": "Fionna",
    "email": "fionna@example.com",
    "phone": "1234567890"
}


let student2 = student1;

console.log(">>> Before Updation: ");
console.log(">>> Std1: ", student1);
console.log(">>> Std2: ", student2);

```

1.4 Manipulate the data for the **student2** object:

student2.email = "fionna.peter@example.com",
student2['hobbies'] = ["Playing", "Cooking"]



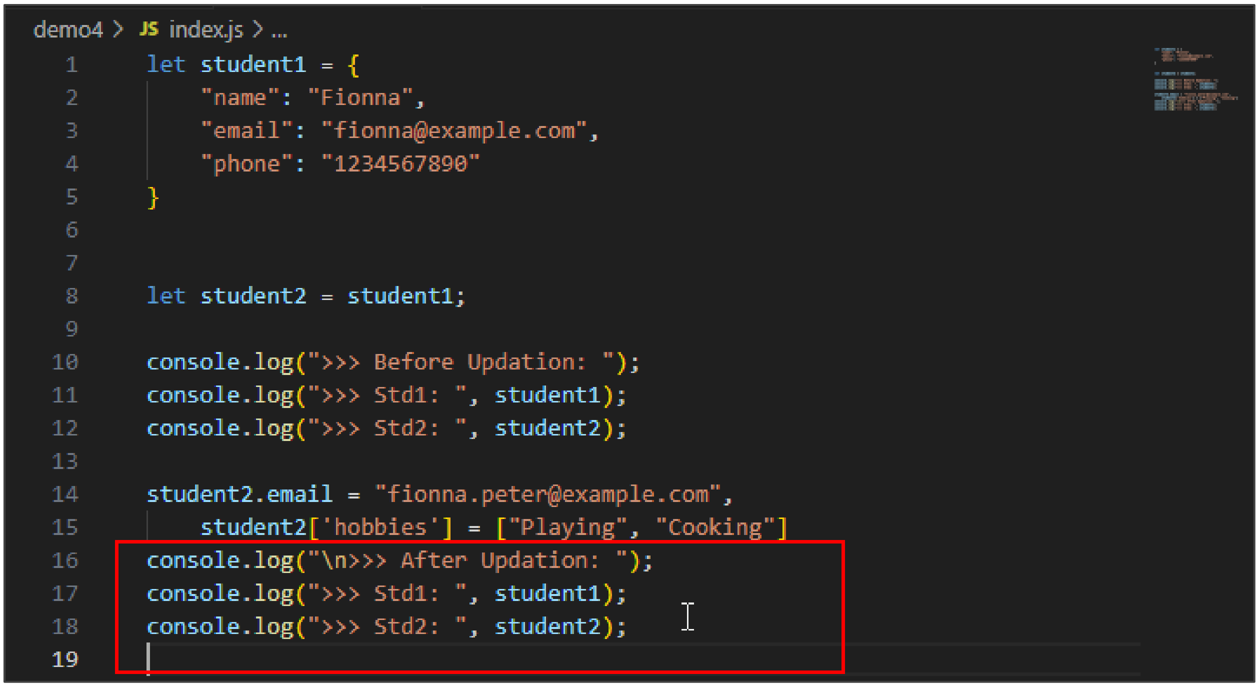
```javascript
let student1 = {
    "name": "Fionna",
    "email": "fionna@example.com",
    "phone": "1234567890"
}


let student2 = student1;

console.log(">>> Before Updation: ");
console.log(">>> Std1: ", student1);
console.log(">>> Std2: ", student2);

student2.email = "fionna.peter@example.com",
    student2['hobbies'] = ["Playing", "Cooking"]

```

1.5 Print the updated details for **student1** and **student2**

console.log("\n>>> After Updation: ");

console.log(">>> Std1: ", student1);

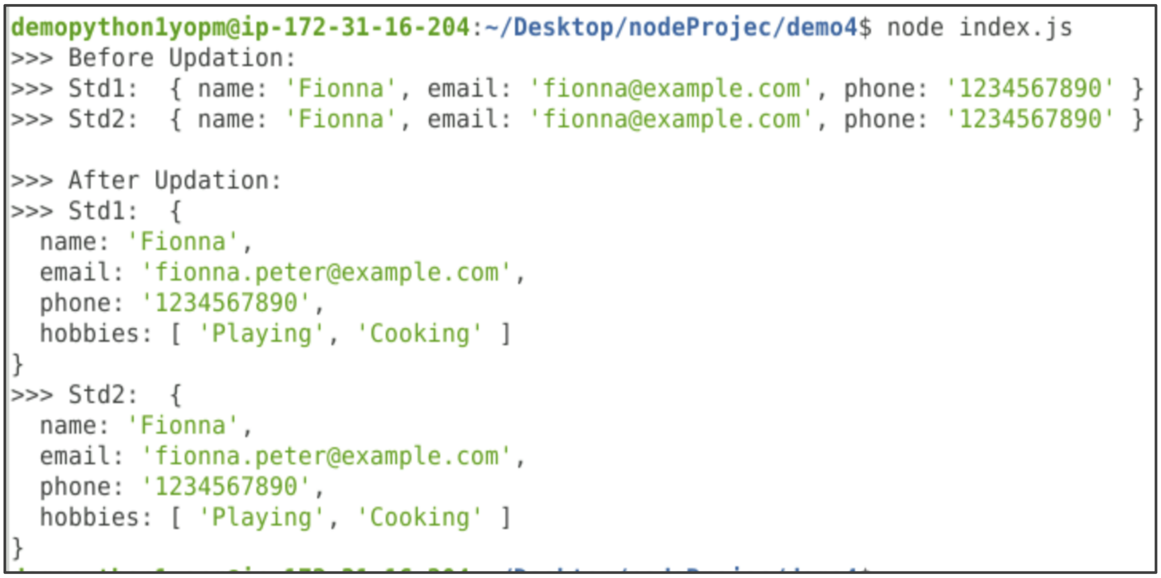
console.log(">>> Std2: ", student2);



```
let student1 = {
    "name": "Fionna",
    "email": "fionna@example.com",
    "phone": "1234567890"
}


let student2 = student1;

console.log(">>> Before Updation: ");
console.log(">>> Std1: ", student1);
console.log(">>> Std2: ", student2);

student2.email = "fionna.peter@example.com",
    student2['hobbies'] = ["Playing", "Cooking"]
console.log("\n>>> After Updation: ");
console.log(">>> Std1: ", student1);
console.log(">>> Std2: ", student2);

```

1.6 Run the following command to get the output:

node index.js



```
demopython1yopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
>>> Before Updation:
>>> Std1:  { name: 'Fionna', email: 'fionna@example.com', phone: '1234567890' }
>>> Std2:  { name: 'Fionna', email: 'fionna@example.com', phone: '1234567890' }

>>> After Updation:
>>> Std1:  {
  name: 'Fionna',
  email: 'fionna.peter@example.com',
  phone: '1234567890',
  hobbies: [ 'Playing', 'Cooking' ]
}
>>> Std2:  {
  name: 'Fionna',
  email: 'fionna.peter@example.com',
  phone: '1234567890',
  hobbies: [ 'Playing', 'Cooking' ]
}
```

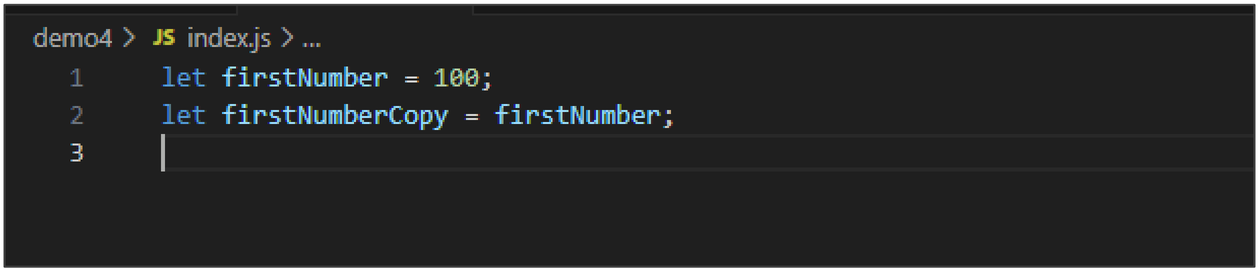
Here, **student2** holds the reference of **student1**. If any changes are made in the **student2** object, they will be reflected in the **student1** object.

Step 2: Create a variable using the pass by value method

2.1 Write the following code to create a variable, then copy its value to another variable:

let firstNumber = 100;

let firstNumberCopy = firstNumber;

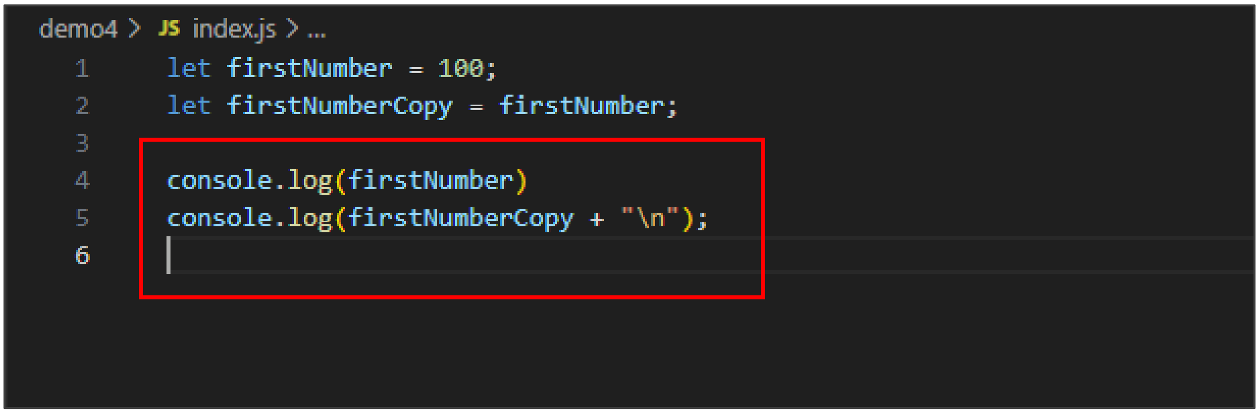


```
let firstNumber = 100;
let firstNumberCopy = firstNumber;

```

2.2 Print the details of both the variables:

console.log(firstNumber)

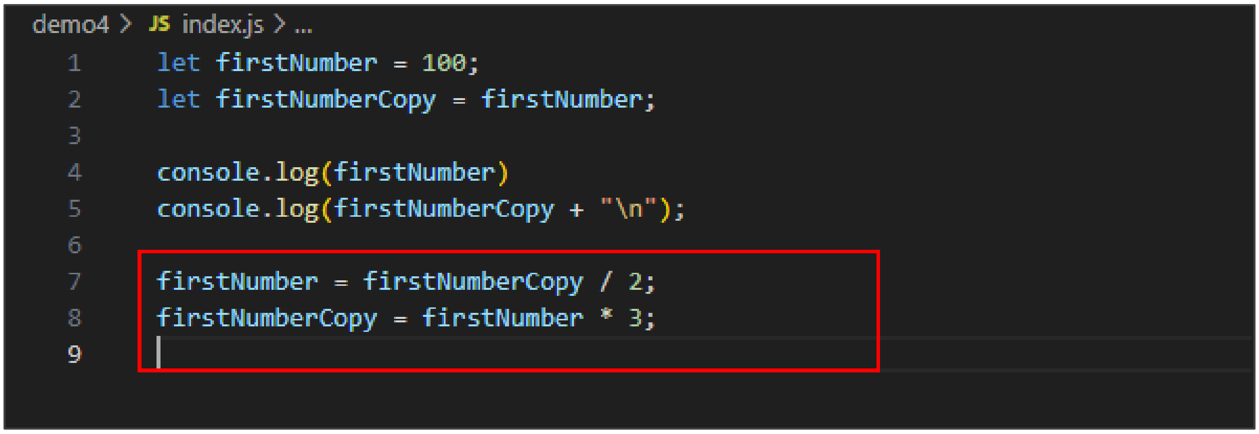
console.log(firstNumberCopy + "\n");



```
let firstNumber = 100;
let firstNumberCopy = firstNumber;

console.log(firstNumber)
console.log(firstNumberCopy + "\n");

```

2.3 Update the values of both the variables using the following arithmetic operation:

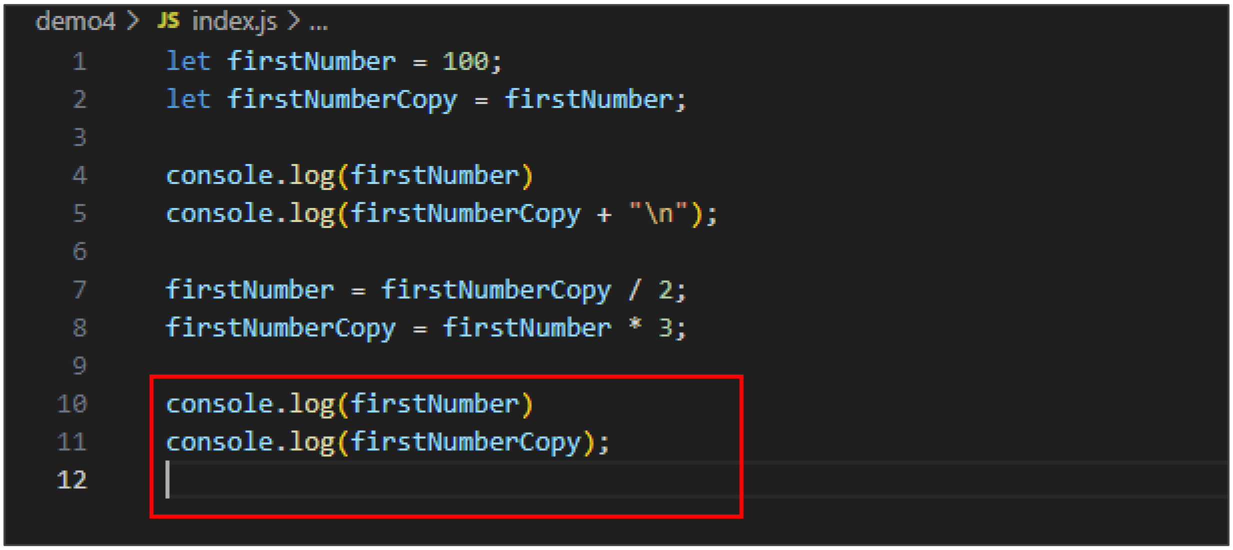
firstNumber = firstNumberCopy / 2;
firstNumberCopy = firstNumber * 3;



```
let firstNumber = 100;
let firstNumberCopy = firstNumber;

console.log(firstNumber)
console.log(firstNumberCopy + "\n");

firstNumber = firstNumberCopy / 2;
firstNumberCopy = firstNumber * 3;

```

2.4 Print the values of both the variables after updating their values

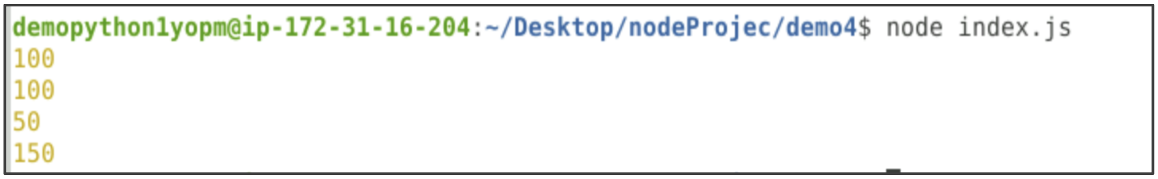
console.log(firstNumber)
console.log(firstNumberCopy);



```
let firstNumber = 100;
let firstNumberCopy = firstNumber;

console.log(firstNumber)
console.log(firstNumberCopy + "\n");

firstNumber = firstNumberCopy / 2;
firstNumberCopy = firstNumber * 3;

console.log(firstNumber)
console.log(firstNumberCopy);

```

2.5 Run the Node.js app using the following command in the terminal to view the output:

node index.js



```
demopython1yopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
100
100
50
150
```

Here, the original and the copied values are independent as they have a different memory space.

By following these steps, you have successfully demonstrated the use of pass by reference and pass by value methods for manipulating the values of the data structures in Node.js.