



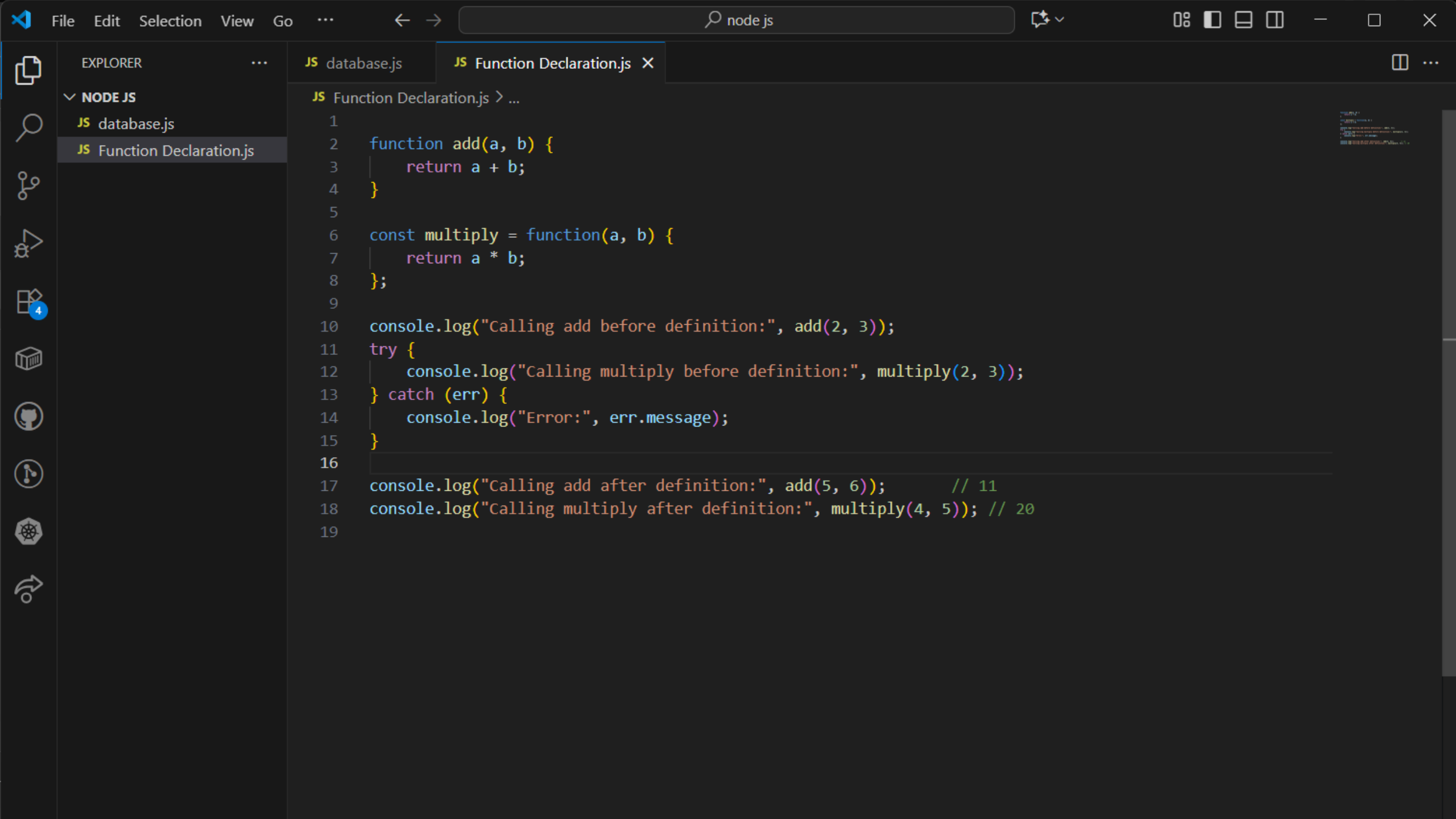
```
1 console.log(a);
2 var a = 10;
3
4
5 try {
6   console.log(b);
7 } catch (err) {
8   console.log("Error with let:", err.message);
9 }
10 let b = 20;
11
12 try {
13   console.log(c);
14 } catch (err) {
15   console.log("Error with const:", err.message);
16 }
17 const c = 30;
18
```

=== Task 1: Hoisting in Variables ===

undefined

Error with let: Cannot access 'b' before initialization

Error with const: Cannot access 'c' before initialization



EXPLORER

...

JS database.js

JS Function Declaration.js X

📄

...

▼ NODE JS

JS database.js

JS Function Declaration.js

JS Function Declaration.js > ...

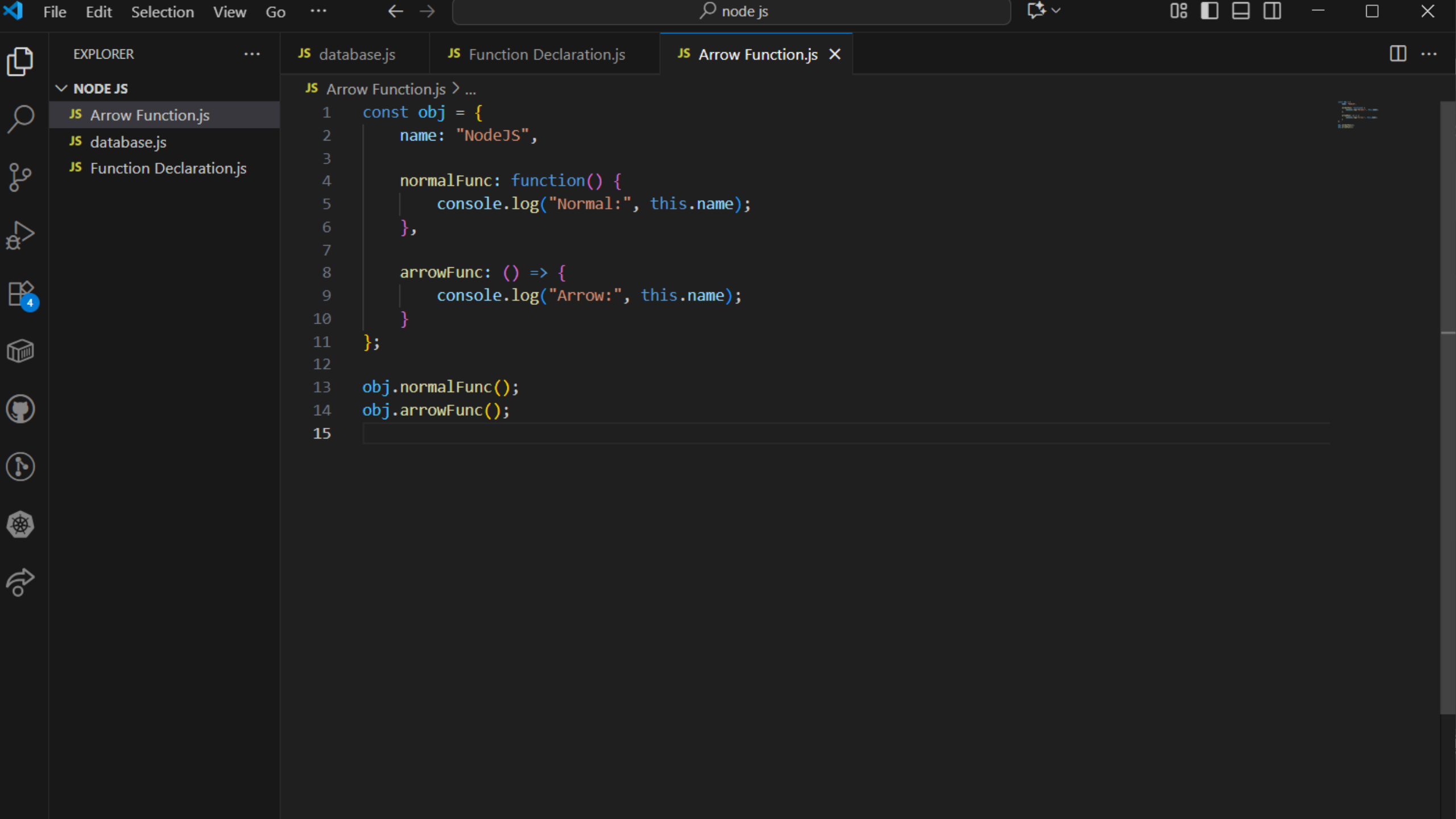
```
1
2 function add(a, b) {
3   return a + b;
4 }
5
6 const multiply = function(a, b) {
7   return a * b;
8 };
9
10 console.log("Calling add before definition:", add(2, 3));
11 try {
12   console.log("Calling multiply before definition:", multiply(2, 3));
13 } catch (err) {
14   console.log("Error:", err.message);
15 }
16
17 console.log("Calling add after definition:", add(5, 6)); // 11
18 console.log("Calling multiply after definition:", multiply(4, 5)); // 20
19
```

Calling add before definition: 5

Error: Cannot access 'multiply' before initialization

Calling add after definition: 11

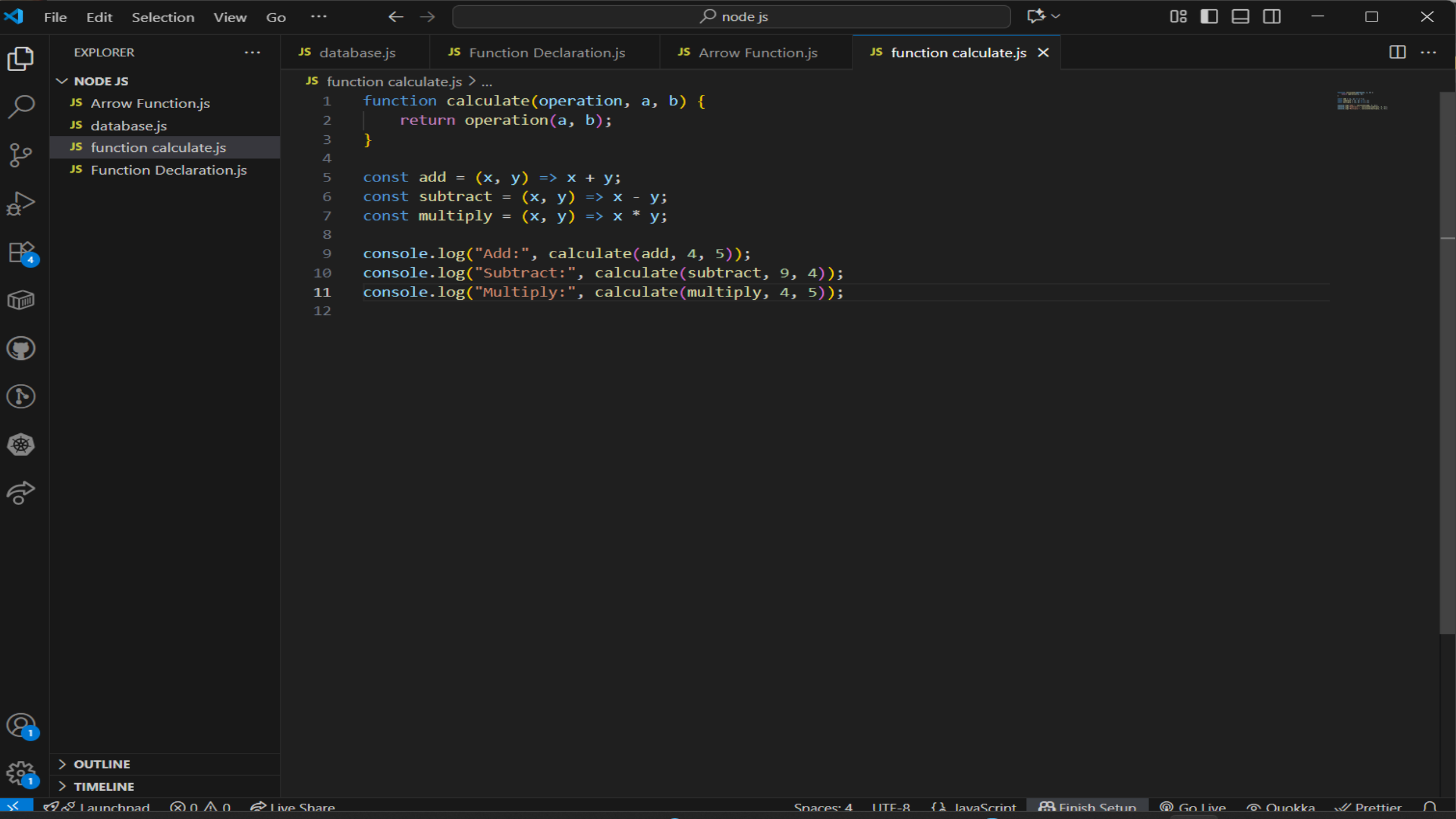
Calling multiply after definition: 20



```
~ node Task3.js
```

```
Normal: NodeJS
```

```
Arrow: undefined
```



```
node.js

JS database.js
JS Function Declaration.js
JS Arrow Function.js
JS function calculate.js X

JS function calculate.js > ...
1  function calculate(operation, a, b) {
2      return operation(a, b);
3  }
4
5  const add = (x, y) => x + y;
6  const subtract = (x, y) => x - y;
7  const multiply = (x, y) => x * y;
8
9  console.log("Add:", calculate(add, 4, 5));
10 console.log("Subtract:", calculate(subtract, 9, 4));
11 console.log("Multiply:", calculate(multiply, 4, 5));
12
```

```
> node Task4.js
```

```
Add: 9
```

```
Subtract: 5
```

```
Multiply: 20
```

```
Divide:
```