

Name: Vince Allen D. Cristal	Date: February 5, 2026
School: LSPU - San Pablo City Campus	Course: On-The-Job Training

1. Challenge 1

Code:

```
function calculateSum(arr) {  
  let sum = 0;  
  for (let i = 0; i < arr.length; i++) {  
    sum += arr[i];  
  }  
  return sum;  
}  
  
let numbers = [1, 2, 3, 4];  
console.log("Sum:", calculateSum(numbers));
```

Output:

10

Screenshot of T-diagram:

Vincent Allen D. Cristol

LSPD - San Pablo City Campus

Challenge 1:

```
1 function calculateSum(arr) {  
2   let sum = 0;  
3   for (let i = 0; i < arr.length; i++) {  
4     sum += arr[i];  
5   }  
6   return sum;  
7 }  
8  
9 let numbers = [1, 2, 3, 4];  
10 console.log("Sum: ", calculateSum(numbers));
```

Output: 10

Variable	Value
numbers	[1, 2, 3, 4]
calculateSum	10

calculateSum	
V	V
sum	0 → 1 → 3 → 6 → 10
i	0 → 1 → 2 → 3

2. Challenge 2

Code:

```
function isEven(num) {  
  if (num % 2 === 0) {  
    return true;  
  } else {  
    return false;  
  }  
}
```

```
console.log(isEven(4));  
console.log(isEven(7));  
console.log(isEven(0));
```

Output:

true

true

true

Screenshot of T-diagram:

Challenge 2

```
1 function isEven(num) {  
2   if (num % 2 === 0) {  
3     return true;  
4   } else {  
5     return false;  
6   }  
7  
8 console.log(isEven(4));  
9 console.log(isEven(7));  
10 console.log(isEven(0));
```

Output :

true

true

true

Variable	Value
num	4, 7, 0
isEven	true

3. Challenge 3

Code:

```
function greet(name) {  
  return "Hello, " + name + "!";  
}  
  
function personalizedGreeting(names) {  
  for (let i = 0; i < names.length; i++) {  
    console.log(greet(names[i]));  
  }  
}  
  
let friends = ["Alice", "Bob", "Charlie"];  
personalizedGreeting(friends);
```

Output:

Hello, Alice!

Hello, Bob!

Hello, Charlie!

Screenshot of T-diagram:

Challenge 3

```
1 function greet(name) {  
2   return "Hullo, " + name + "!";  
3 }  
4  
5 function personalizedGreeting(names) {  
6   for (let i=0; i < names.length; i++) {  
7     console.log(greet(names[i]));  
8   }  
9 }  
10  
11 let friends = ["Alice", "Bob", "Charlie"];  
12 personalizedGreeting(friends);
```

Output:

Variable	Value
friends	["Alice", "Bob", "Charlie"]
personalizedGreeting	Hullo, Alice!
	Hullo, Bob!
	Hullo, Charlie!

personalizedGreeting	
V	V
names	["Alice", "Bob", "Charlie"]
i	0 → 1 → 2 → 3

4. Challenge 4

Code:

```
function reverseArray(arr) {  
  let reversed = [];  
  for (let i = arr.length - 1; i >= 0; i--) {  
    reversed.push(arr[i]);  
  }  
  return reversed;  
}  
  
let originalArray = [10, 20, 30];  
console.log(reverseArray(originalArray));  
console.log(originalArray);
```

Output:

[30, 20, 10]

[10, 20, 30]

Screenshot of T-diagram:

Challenge 4

```
1 function reverseArray (arr) {  
2   let reversed = []  
3   for (let i = arr.length - 1; i >= 0; i--) {  
4     reversed.push(arr[i]);  
5   }  
6   return reversed;  
7 }  
8  
9 let originalArray = [10, 20, 30]  
10 console.log(reverseArray(originalArray));  
11 console.log(originalArray);
```

Output:

[20, 20, 10]

[10, 20, 30]

Variable

Value

originalArray

[10, 20, 30]

reverseArray

[20, 20, 10]

reverseArray

V

V

reversed

[] → [30] → [30, 20] →

[30, 20, 10]

i

2 → 1 → 0

5. Challenge

Code:

```
function multiplyMatrix(matrix) {  
  for (let i = 0; i < matrix.length; i++) {  
    for (let j = 0; j < matrix[i].length; j++) {  
      matrix[i][j] *= 2;  
    }  
  }  
  return matrix;  
}
```

```
let matrix = [  
  [1, 2],  
  [3, 4],  
];  
  
console.log(multiplyMatrix(matrix));
```

Output:

```
[[2, 4], [6, 8]]
```

Screenshot of T-diagram:

Challenge 5

```

1 function multiplyMatrix(matrix)
2   for (let i = 0; i < matrix.length; i++) {
3     for (let j = 0; j < matrix[i].length; j++) {
4       matrix[i][j] *= 2
5     }
6   }
7   return matrix;
8
9 let matrix = [
10   [1, 2],
11   [3, 4]
12 ];
13 console.log(multiplyMatrix(matrix));

```

Output

[[2, 4], [6, 8]]

Variable

Value

matrix

[[1, 2], [3, 4]]

multiplyMatrix

[[2, 4], [6, 8]]

Matrix Based T-Diagram

[[2, 4], [6, 8]]

[[2, 4], [3, 4]]

[[2, 4], [6, 4]]

[[2, 4], [6, 8]]

multiplyMatrix

✓

i

j

✓

0 → 1

0 → 1 → 0 → 1