Lesson 02 Demo 12

Implementing Stacks Using Arrays

Objective: To implement a stack using arrays in JavaScript, covering essential operations like push, pop, peek, and displaying the stack's contents

Tools required: Visual Studio Code and Node.js

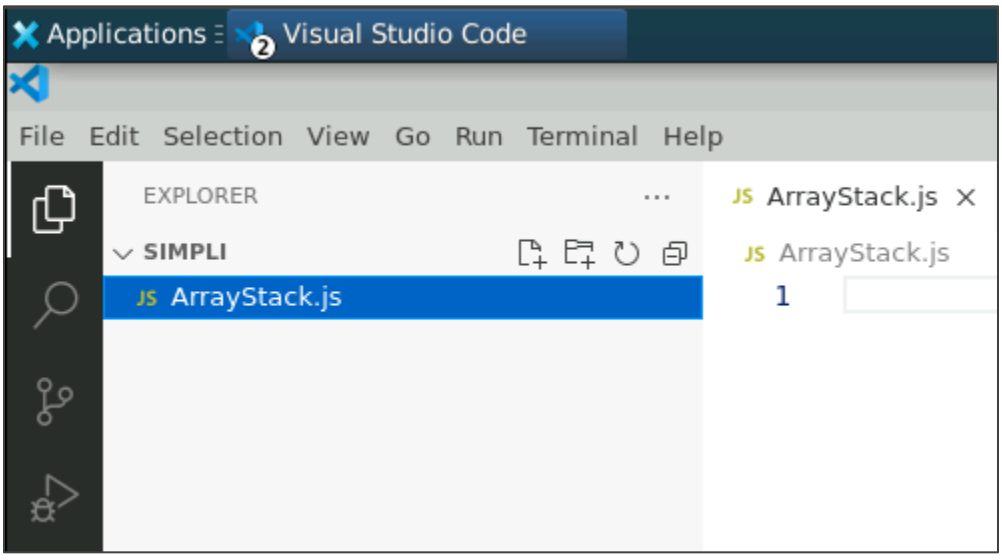
Prerequisites: Basic understanding of arrays in JavaScript

Steps to be followed:

1. Create and execute the JS file

Step 1: Create and execute the JS file

1.1 Open the Visual Studio Code editor and create a JavaScript file named **ArrayStack.js**



1.2 Write the code given below in the **ArrayStack.js** file:

```
// Implementing a Stack Using an Array
class ArrayStack {
  constructor() {
    this.stack = [];
  }

  // Push operation
  push(item) {
    this.stack.push(item);
  }

  // Pop operation
  pop() {
    if (this.stack.length === 0) {
      return "Stack is empty";
    }
    return this.stack.pop();
  }

  // Top operation (Peek)
  top() {
    return this.stack[this.stack.length - 1];
  }

  // Check if the stack is empty
  isEmpty() {
    return this.stack.length === 0;
  }

  // Display the stack
  display() {
    console.log("Stack:", this.stack);
  }
}
// Using the ArrayStack
let stack = new ArrayStack();
// Pushing items
stack.push('Apple');
stack.push('Banana');
stack.push('Cherry');
```

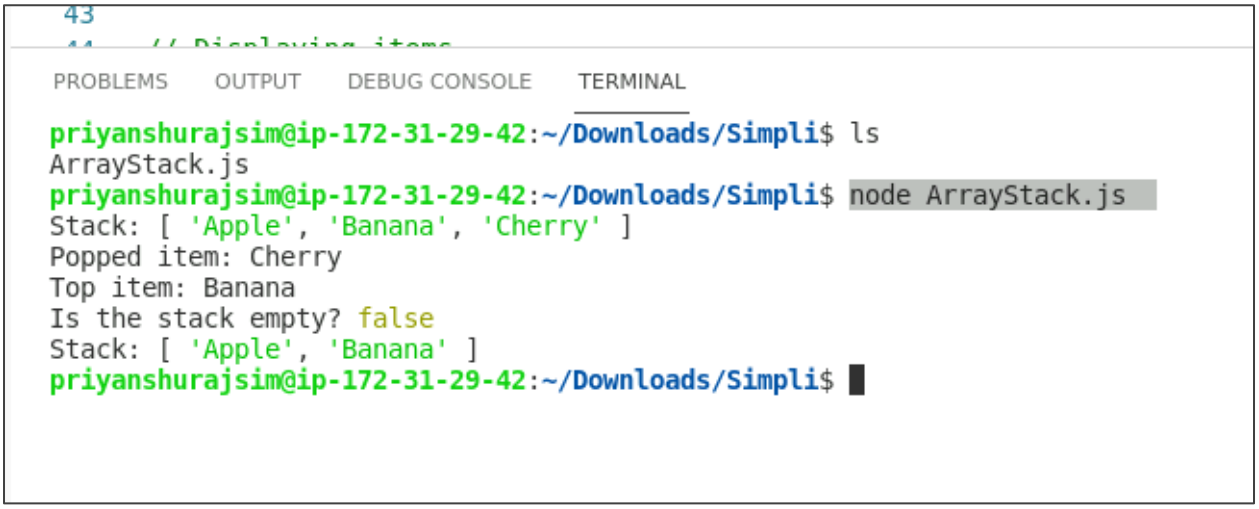
```
// Displaying items
stack.display();

// Popping an item
console.log("Popped item:", stack.pop());

// Displaying the top item
console.log("Top item:", stack.top());

// Checking if the stack is empty
console.log("Is the stack empty?", stack.isEmpty());

// Displaying the stack
stack.display();
```

```
1  // Implementing a Stack Using an Array
2  class ArrayStack {
3      constructor() {
4          this.stack = [];
5      }
6
7      // Push operation
8      push(item) {
9          this.stack.push(item);
10     }
11
12     // Pop operation
13     pop() {
14         if (this.stack.length === 0) {
15             return "Stack is empty";
16         }
17         return this.stack.pop();
18     }
19
20     // Top operation (Peek)
21     top() {
22         return this.stack[this.stack.length - 1];
23     }
24
```

```
25      // Check if the stack is empty
26      isEmpty() {
27          return this.stack.length === 0;
28      }
29
30      // Display the stack
31      display() {
32          console.log("Stack:", this.stack);
33      }
34  }
35
36  // Using the ArrayStack
37  let stack = new ArrayStack();
38
39  // Pushing items
40  stack.push('Apple');
41  stack.push('Banana');
42  stack.push('Cherry');
43
44  // Displaying items
45  stack.display();
46
```

```
46
47  // Popping an item
48  console.log("Popped item:", stack.pop());
49
50  // Displaying the top item
51  console.log("Top item:", stack.top());
52
53  // Checking if the stack is empty
54  console.log("Is the stack empty?", stack.isEmpty());
55
56  // Displaying the stack
57  stack.display();
58
```

1.3 Save the file and execute it in the terminal using the command given below:
node ArrayStack.js



This example illustrates the creating a stack with an array in JavaScript.

By following these steps, you have successfully managed a stack with arrays in JavaScript, enhancing your data structure manipulation skills.