

# Lesson 02 Demo 10

## Implementing CRUD Operations on a Stack

**Objective:** To implement CRUD operations on a stack, showcasing how to push, pop, peek, display, and clear elements within a stack structure, enhancing your ability to manipulate data structures and apply fundamental programming concepts in practical scenarios

**Tools required:** Visual Studio Code

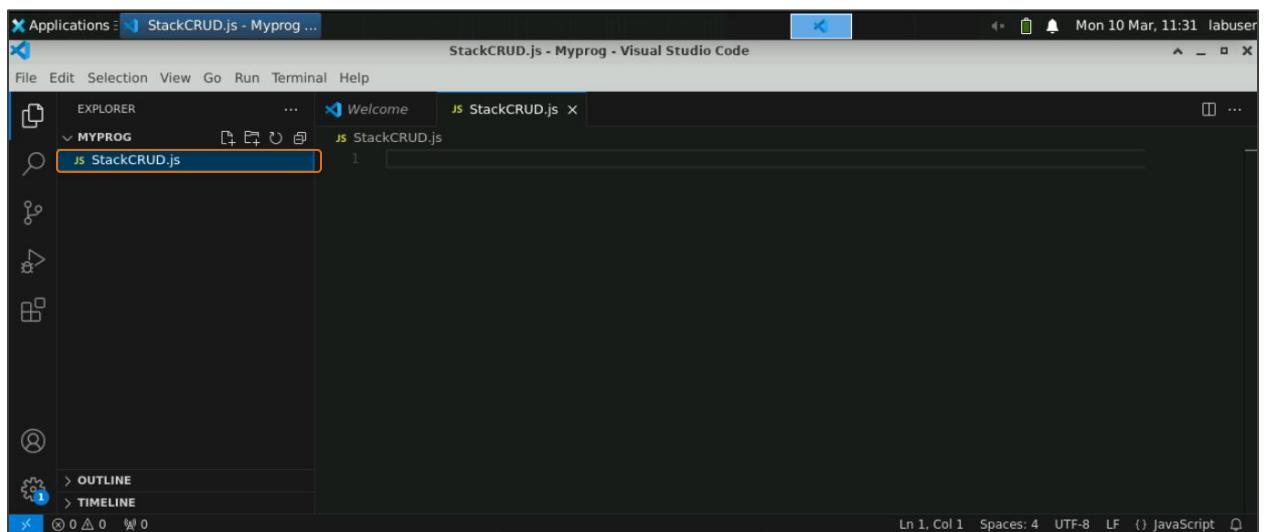
**Prerequisites:** A basic understanding of data structures and JavaScript

Steps to be followed:

1. Create a JavaScript file and execute it

### Step 1: Create a JavaScript file and execute it

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



1.2 Add the following code to the file:

```
// Implementing CRUD Operations on a Stack

class Stack {
    constructor() {
        this.items = [];
    }

    // Push operation
    push(element) {
        this.items.push(element);
    }

    // Pop operation
    pop() {
        if (this.items.length === 0) {
            return "Underflow";
        }
        return this.items.pop();
    }

    // Peek operation
    peek() {
        return this.items[this.items.length - 1];
    }

    // Display the Stack
    display() {
        console.log("Stack elements:", this.items);
    }

    // Clear the Stack
    clear() {
        this.items = [];
        console.log("Stack cleared.");
    }
}

// Creating a stack
let myStack = new Stack();
```

```
// Pushing elements onto the stack
myStack.push(10);
myStack.push(20);
myStack.push(30);

// Displaying the stack
myStack.display();

// Popping an element from the stack
let poppedElement = myStack.pop();
console.log("Popped element:", poppedElement);

// Displaying the stack after popping
myStack.display();

// Peeking into the stack
let topElement = myStack.peek();
console.log("Top element:", topElement);

// Clearing the stack
myStack.clear();
myStack.display();
```

StackCRUD.js - Myprog - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER JS StackCRUD.js

Welcome JS StackCRUD.js

```
JS StackCRUD.js > ...
1 // Implementing CRUD Operations on a Stack
2
3 class Stack {
4     constructor() {
5         this.items = [];
6     }
7
8     // Push operation
9     push(element) {
10        this.items.push(element);
11    }
12
13     // Pop operation
14     pop() {
15        if (this.items.length === 0) {
16            return "Underflow";
17        }
18        return this.items.pop();
19    }
}
```

Ln 87, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

StackCRUD.js - Myprog - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER JS StackCRUD.js

Welcome JS StackCRUD.js

```
JS StackCRUD.js > Stack > pop
3 class Stack {
21
22     // Peek operation
23     peek() {
24         return this.items[this.items.length - 1];
25     }
26
27     // Display the Stack
28     display() {
29         console.log("Stack elements:", this.items);
30     }
31
32     // Clear the Stack
33     clear() {
34         this.items = [];
35         console.log("Stack cleared.");
36     }
37 }
```

Ln 19, Col 6 Spaces: 4 UTF-8 LF {} JavaScript

StackCRUD.js - Myprog - Visual Studio Code

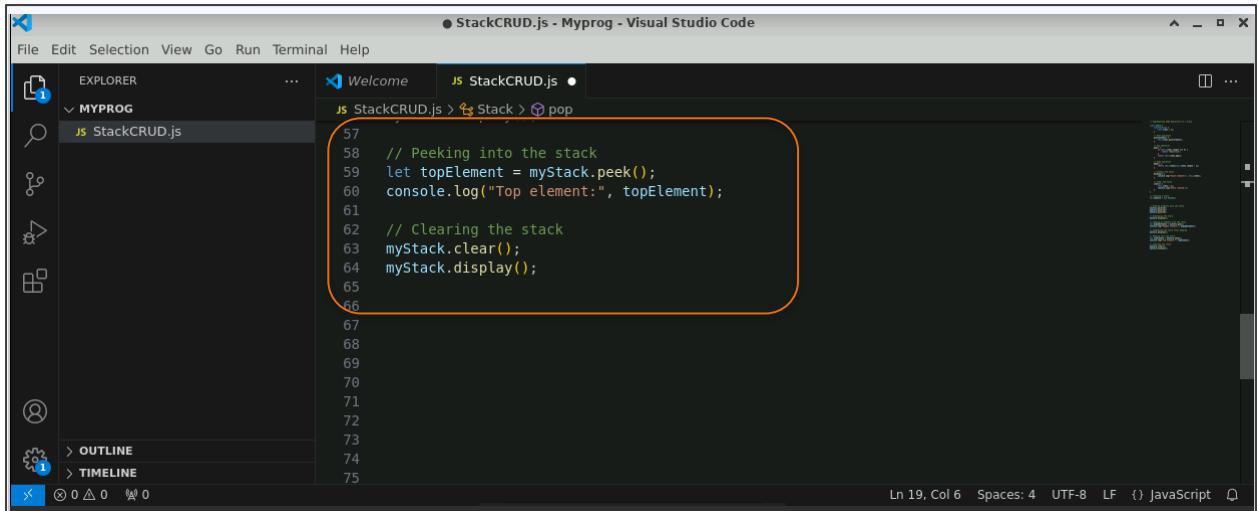
File Edit Selection View Go Run Terminal Help

EXPLORER JS StackCRUD.js

Welcome JS StackCRUD.js

```
JS StackCRUD.js > Stack > pop
38 // Creating a stack
39 let myStack = new Stack();
40
41
42 // Pushing elements onto the stack
43 myStack.push(10);
44 myStack.push(20);
45 myStack.push(30);
46
47 // Displaying the stack
48 myStack.display();
49
50
51 // Popping an element from the stack
52 let poppedElement = myStack.pop();
53 console.log("Popped element:", poppedElement);
54
55 // Displaying the stack after popping
56 myStack.display();
```

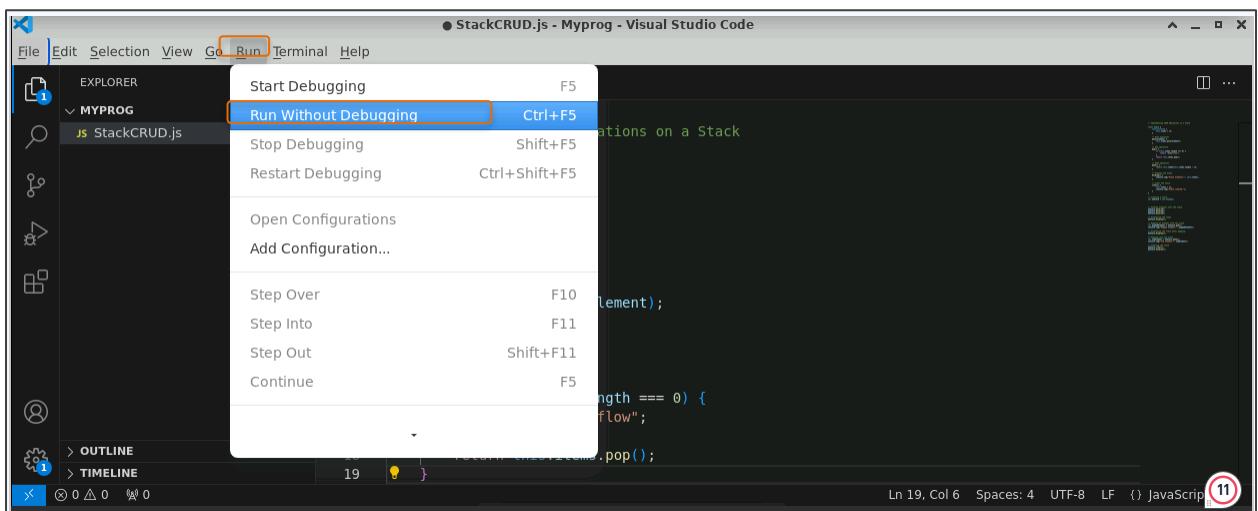
Ln 19, Col 6 Spaces: 4 UTF-8 LF {} JavaScript



```
● StackCRUD.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER ... JS StackCRUD.js ●
MYPROG StackCRUD.js
StackCRUD.js > Stack > pop
57
58 // Peeking into the stack
59 let topElement = myStack.peek();
60 console.log("Top element:", topElement);
61
62 // Clearing the stack
63 myStack.clear();
64 myStack.display();
65
66
67
68
69
70
71
72
73
74
75
```

Ln 19, Col 6 Spaces: 4 UTF-8 LF {} JavaScript

1.3 Click **Run** and then **Run Without Debugging**. Select **Node.js** to check the output in the **DEBUG CONSOLE**.



A screenshot of the Visual Studio Code interface. The title bar says "StackCRUD.js - Myprog - Visual Studio Code". The left sidebar shows an "EXPLORER" view with a folder "MYPROG" containing a file "StackCRUD.js". A "SELECT DEBUGGER" dropdown menu is open, with "Node.js" highlighted. Below it are options for "VS Code Extension Development", "Web App (Chrome)", and "Web App (Edge)". The main code editor area contains the following JavaScript code:

```
6 }  
7 // Push operation  
8 push(element) {  
9   this.items.push(element);  
10 }  
11  
12 // Pop operation  
13 pop() {  
14   if (this.items.length === 0) {  
15     return "Underflow";  
16   }  
17   return this.items.pop();  
18 }  
19 }
```

The status bar at the bottom right shows "Ln 19, Col 6 Spaces: 4 UTF-8 LF {} JavaScript".

#### 1.4 View the output in the **DEBUG CONSOLE** as shown below:

A screenshot of the Visual Studio Code interface. The title bar says "StackCRUD.js - Myprog - Visual Studio Code". The left sidebar shows an "EXPLORER" view with a folder "MYPROG" containing a file "StackCRUD.js". The "DEBUG CONSOLE" tab is selected in the bottom navigation bar. The console output shows the execution of a script and its results:

```
1 // Implementing CRUD Operations on a Stack  
2  
3 class Stack {  
4   constructor() {  
5     this.items = [];  
6   }  
7  
8   push(element) {  
9     this.items.push(element);  
10 }  
11  
12 pop() {  
13   if (this.items.length === 0) {  
14     return "Underflow";  
15   }  
16   return this.items.pop();  
17 }  
18  
19 }  
20  
21 const stack = new Stack();  
22 stack.push(10);  
23 stack.push(20);  
24 stack.push(30);  
25  
26 console.log("Stack elements: ", stack.items);  
27  
28 const poppedElement = stack.pop();  
29  
30 console.log("Popped element: ", poppedElement);  
31  
32 console.log("Stack elements: ", stack.items);  
33  
34 stack.pop();  
35  
36 console.log("Top element: ", stack.items[stack.items.length - 1]);  
37  
38 stack.clear();  
39  
40 console.log("Stack cleared.");  
41  
42 console.log("Stack elements: ", stack.items);  
43  
44
```

The status bar at the bottom right shows "Ln 19, Col 6 Spaces: 4 UTF-8 LF {} JavaScript".

**Note:** This example illustrates CRUD operations on a stack, including pushing, popping, peeking, displaying, and clearing elements.

By following these steps, you have successfully implemented various CRUD operations a stack in JavaScript, enhancing your ability to manipulate data structures and apply fundamental programming concepts in practical scenarios.