

Lesson 02 Demo 08

Implement CRUD Operations on a Doubly Circular Linked List

Objective: To create a doubly circular linked list in JavaScript with CRUD functionalities such as node addition, traversal, value modification, and node deletion

Tools required: Visual Studio Code (VS Code) and JavaScript

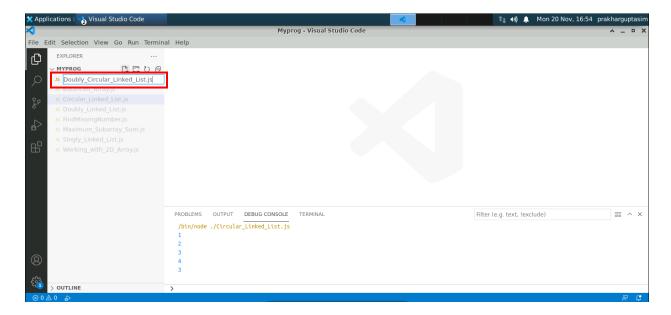
Prerequisites: Perform demo 01 of lesson 02

Steps to be followed:

1. Create and execute the JS file

Step 1: Create and execute the JS file

1.1 Create a JavaScript file named **Doubly_Circular_Linked_List.js** as shown below:





1.2 Write the code in the file created in step 1.1 as shown below:

```
class ListNode {
  constructor(data) {
    this.data = data;
    this.next = null;
    this.prev = null;
 }
}
class DoublyCircularLinkedList {
  constructor() {
    this.head = null;
 }
 // Create: Add a new node to the list
  add(data) {
    const newNode = new ListNode(data);
    if (!this.head) {
      this.head = newNode;
      newNode.next = newNode;
      newNode.prev = newNode;
    } else {
      newNode.prev = this.head.prev;
      newNode.next = this.head;
      this.head.prev.next = newNode;
      this.head.prev = newNode;
    }
 }
  // Read: Traverse and display elements of the list
  read() {
    if (!this.head) {
      return;
    }
    let current = this.head;
```



```
do {
    console.log(current.data);
    current = current.next;
  } while (current !== this.head);
}
// Update: Modify the value of a node at a given position
update(position, data) {
  if (!this.head) {
    return;
  }
  let current = this.head;
  let count = 0;
  do {
    if (count === position) {
      current.data = data;
      return;
    }
    current = current.next;
    count++;
  } while (current !== this.head);
  console.log("Position not found");
}
// Delete: Remove a node from the list at a specified position
delete(position) {
  if (!this.head) {
    return;
  }
  if (this.head.next === this.head) {
    if (position === 0) {
      this.head = null;
    return;
  }
```

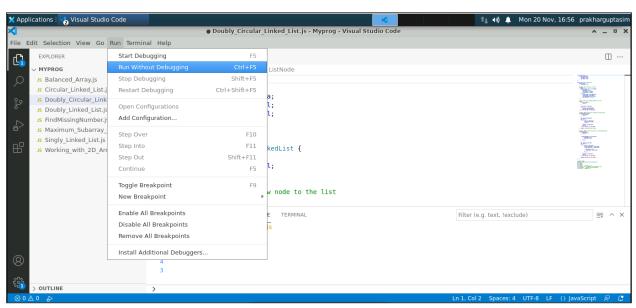


```
let current = this.head;
    let count = 0;
    do {
      if (count === position) {
         current.prev.next = current.next;
         current.next.prev = current.prev;
         if (position === 0) {
           this.head = current.next;
         }
         return;
      }
      current = current.next;
      count++;
    } while (current !== this.head);
    console.log("Position not found");
  }
}
// Example usage
const list = new DoublyCircularLinkedList();
list.add(1);
list.add(2);
list.add(3);
list.read(); // Displays 1, 2, 3
list.update(1, 4); // Updates the second element to 4
list.delete(0); // Deletes the first element
list.read(); // Displays 4, 3
```



```
🗶 Applications 🗄 🌏 Visual Studio Code
                                                                    Doubly_Circular_Linked_List.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
                                               JS Doubly_Circular_Linked_List.js ×
 Ф
         MYPROG
                                                JS Doubly_Circular_Linked_List.js > 😝 DoublyCircularLinkedList
                                                        class ListNode {
         JS Balanced Array.js
                                                             constructor(data) {
         JS Circular Linked List.js
                                                                   this.data = data;
this.next = null;
this.prev = null;
        JS Doubly_Circular_Linked_List.js
        JS Doubly_Linked_List.js
        JS FindMissingNumber.js
         JS Maximum_Subarray_Sum.js
         JS Singly_Linked_List.js
                                                        class DoublyCircularLinkedList {
        JS Working_with_2D_Array.js
                                                  10
11
                                                              constructor() {
   this.head = null;
                                                  12
13
14
15
16
17
18
19
20
21
22
23
24
                                                              // Create: Add a new node to the list
                                                              add(data) {
    const newNode = new ListNode(data);
                                                                   if (!this.head) {
                                                                        this head = newNode;
                                                                        newNode.next = newNode;
newNode.prev = newNode;
                                                                   } else {
   newNode.prev = this.head.prev;
                                                                         newNode.next = this.head;
this.head.prev.next = newNode;
                                                                        this head.prev = newNode;
```

1.3 Save the code and click on **Run->Run Without Debugging->Node.js** to check the output in the debug console





```
🕻 Applications 🗄 🌏 Visual Studio Code
                                                           Doubly_Circular_Linked_List.js - Myprog - Visual Studio Code
      EXPLORER
                                   ... Js Doubly Select environment
      ✓ MYPROG
                                          JS Doubly Chrome
      JS Balanced Array.js
       JS Circular Linked List.js
      JS Doubly_Circular_Linked_List.js
       JS Doubly_Linked_List.js
                                                    Install an extension for JavaScript...
       JS FindMissingNumber.js
      Js Maximum_Subarray_Sum.js
      JS Singly_Linked_List.js
                                                 class DoublyCircularLinkedList {
       JS Working_with_2D_Array.js
                                                     constructor() {
                                                      // Create: Add a new node to the list
                                                      add(data) {
                                           PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
                                                                                                                                      Filter (e.g. text, !exclude)
                                            /bin/node ./Circular_Linked_List.js
      OUTLINE
```

Now you can see the output in the debug console as shown below:

```
Doubly Circular Linked List.js - Myprog - Visual Studio Code
       EXPLORER
                                  ··· Js Doubly_Circular_Linked_List.js ×
                                                                                                                                                                                    □ ...
Ð
                                          JS Doubly Circular Linked List.is > % ListNode
       JS Balanced_Array.js
                                                       constructor(data) {
       JS Circular_Linked_List.js
                                                           this.data = data;
this.next = null;
       Js Doubly_Circular_Linked_List.js
       JS Doubly_Linked_List.js
                                                            this.prev = null;
       JS FindMissingNumber.js
       Js Maximum_Subarray_Sum.js
       Js Singly_Linked_List.js
       JS Working_with_2D_Array.js
                                                 class DoublyCircularLinkedList {
                                                      constructor() {
   this.head = null;
                                            11
                                            13
                                                       // Create: Add a new node to the list
                                            15
                                                       add(data) {
                                           PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
                                                                                                                                        Filter (e.g. text, !exclude)
                                                                                                                                                                                  > OUTLINE
```

By following the above steps, you have successfully performed the **CRUD** operations on a double circular linked list. Here, the **add()** method adds a new node at the end of the list, **read()** method traverses and prints the list, **update()** method changes the value at a given position, and **delete()** method removes a node at a specified position