

Lesson 02 Demo 07

Implement CRUD Operations on a Circular Linked List

Objective: To create a 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

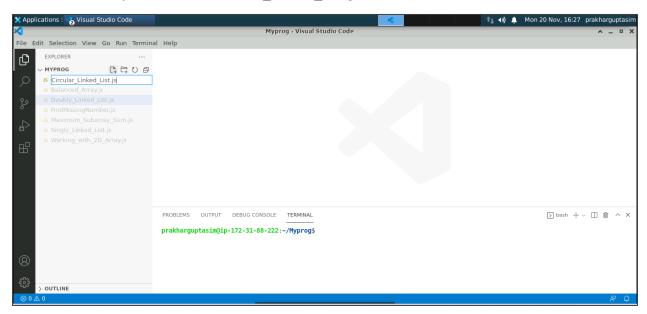
Prerequisites: Perform demo 01 of lesson 02

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 Circular_Linked_List.js as shown below:





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

```
class ListNode {
  constructor(data) {
    this.data = data;
    this.next = null;
  }
}
class CircularLinkedList {
  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 = this.head;
    } else {
      let current = this.head;
      while (current.next !== this.head) {
         current = current.next;
      }
      current.next = newNode;
      newNode.next = this.head;
    }
  }
  // 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 (position === 0) {
    if (this.head.next === this.head) {
      this.head = null;
    } else {
      let current = this.head;
       while (current.next !== this.head) {
         current = current.next;
```



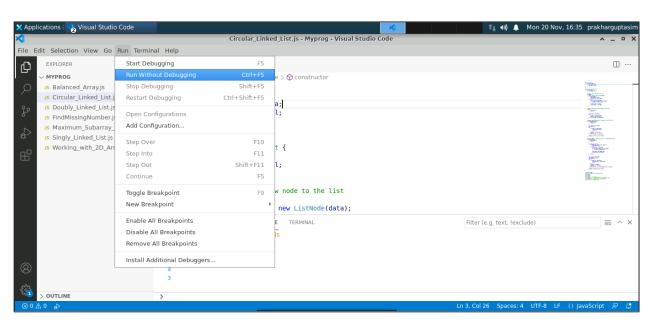
```
}
         this.head = this.head.next;
         current.next = this.head;
      }
      return;
    }
    let current = this.head;
    let previous = null;
    let count = 0;
    do {
      if (count === position) {
         previous.next = current.next;
         return;
      }
      previous = current;
      current = current.next;
      count++;
    } while (current !== this.head && current !== null);
    console.log("Position not found");
 }
}
// Example usage
const list = new CircularLinkedList();
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
File Edit Selection View Go Run Terminal Help
        EXPLORER
                                     ... Js Circular_Linked_List.js 

                                                                                                                                                                                         □ ...
 G
         MYPROG
                                             JS Circular_Linked_List.js >
         JS Balanced_Array.js
                                               1 class ListNode (
        JS Circular_Linked_List.js
                                                              this.data = data;
this.next = null;
        JS Doubly Linked List.js
        JS FindMissingNumber.js
        JS Maximum_Subarray_Sum.js
        JS Singly_Linked_List.js
                                                    class CircularLinkedList {
   constructor() {
      this.head = null;
}
         JS Working_with_2D_Array.js
                                              10
11
                                              12
13
14
15
                                                         // Create: Add a new node to the list
                                                         add(data) {
   const newNode = new ListNode(data);
                                             PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
                                                                                                                                                                      prakharguptasim@ip-172-31-88-222:~/Myprog$
                                                                                                                                 (i) There is an available update.
       > OUTLINE
```

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





```
X Applications = V2 Visual Studio Code
                                                           Circular Linked List.js - Myprog - Visual Studio Code
                            ... Js Circula Select environment
                         다 타 라 이 의 S Circula Chrome
       ✓ MYPROG
       JS Balanced Array.is
                                         1 C Edge: Launch
       Js 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
                                                  constructor() {
    this.head = null;
}
                                         8 class CircularLinkedList {
       JS Working_with_2D_Array.js
                                         11
                                         12
                                                   // Create: Add a new node to the list
                                                   add(data) {
                                                   const newNode = new ListNode(data);
                                         PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

    bash + ∨ □ □ □ ^ ×
                                        prakharguptasim@ip-172-31-88-222:~/Myprog$
      > OUTLINE
```

• You can see the output in the debug console as shown below:

```
X Applications : Visual Studio Code
                                                                                                                                   👊 🜒 🔔 Mon 20 Nov, 16:37 prakharguptas
File Edit Selection View Go Run Terminal Help
       EXPLORER
                         ··· JS Circular Linked List.js X
                                                                                                                                                                     □ ..
 C
        JS Balanced_Array.js
                                         1 class ListNode {
                                              constructor(data) {
this data
       JS Circular_Linked_List.js
                                                       this data = data:
        JS Doubly_Linked_List.js
                                                       this.next = null;
        JS FindMissingNumber.js
        Js Maximum_Subarray_Sum.js
        JS Singly_Linked_List.js
                                          8 class CircularLinkedList {
        JS Working_with_2D_Array.js
                                                 constructor() {
                                         11
                                                  // Create: Add a new node to the list
                                         13
                                                  add(data) {
    const newNode = new ListNode(data);
                                         15
                                         PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
                                                                                                                            Filter (e.g. text, !exclude)
                                          /bin/node ./Circular_Linked_List.js
                                       > Please start a debug session to evaluate expression
```

By following the above steps, you have successfully performed the **CRUD** operations on a 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.