**Code: -**

class Node:

    def \_\_init\_\_(self, data):

        self.data = data  # Store the data of the node

        self.next = None  # Initialize the next pointer as None

class SinglyLinkedList:

    def \_\_init\_\_(self):

        self.head = None  # Initialize the list with no elements (empty list)

    def append(self, data):

        new\_node = Node(data)  # Create a new node

        if not self.head:  # If the list is empty, make new node the head

            self.head = new\_node

            return

        last = self.head

        while last.next:  # Traverse to the last node

            last = last.next

        last.next = new\_node  # Append the new node at the end

    def prepend(self, data):

        new\_node = Node(data)  # Create a new node

        new\_node.next = self.head  # Point new node to the current head

        self.head = new\_node  # Make new node the head of the list

    def delete(self, key):

        current = self.head

        if current and current.data == key:  # If the node to delete is the head

            self.head = current.next  # Move the head to the next node

            current = None

            return

        prev = None

        while current and current.data != key:  # Traverse the list

            prev = current

            current = current.next

        if current is None:  # If the node is not found

            print(f"Node with value {key} not found.")

            return

        prev.next = current.next  # Unlink the node from the list

        current = None

    def search(self, key):

        current = self.head

        while current:

            if current.data == key:  # If the node is found

                return True

            current = current.next

        return False  # Return False if the node is not found

    def display(self):

        if not self.head:

            print("The list is empty.")

            return

        current = self.head

        while current:

            print(current.data, end="   - > ")  # Print the node data

            current = current.next

        print("None")  # End of the list

# Example usage of the SinglyLinkedList class

if \_\_name\_\_ == "\_\_main\_\_":

    sll = SinglyLinkedList()  # Create an instance of SinglyLinkedList

    sll.append(10)

    sll.append(20)

    sll.append(30)

    sll.append(40)

    print("List after appending 10, 20, 30, 40:")

    sll.display()  # Expected Output: 10   - > 20   - > 30   - > 40   - > None

    sll.prepend(5)

    print("List after prepending 5:")

    sll.display()  # Expected Output: 5   - > 10   - > 20   - > 30   - > 40   - > None

    print("Searching for 20 in the list:", sll.search(20))  # Expected Output: True

    print("Searching for 100 in the list:", sll.search(100))  # Expected Output: False

    sll.delete(20)

    print("List after deleting node with value 20:")

    sll.display()  # Expected Output: 5   - > 10   - > 30   - > 40   - > None

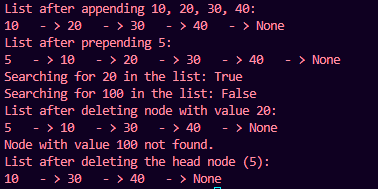
    sll.delete(100)  # Expected Output: Node with value 100 not found.

    sll.delete(5)

    print("List after deleting the head node (5):")

    sll.display()  # Expected Output: 10   - > 30   - > 40   - > None

**OUTPUT:**

****