NAME: IGWENAGU CHINWEIKE OBIORA

REGISTRATION NUMBER: 2023/257473

DEPARTMENT: COMPUTER SCIENCE

TITLE: COS 232 ASSIGNMENT

Full code on my [github](https://github.com/Nweike-Igwenagu/COS-232-Assignment.git).

https://github.com/Nweike-Igwenagu/COS-232-Assignment.git

1. Stack Implementation

Code:

```
public class Stack {
   private final int[] implArr;
   int length;
   private int top = -1;

   public Stack(int length) {
       this.length = length;
       this.implArr = new int[length];
   }

   int getTop() {return this.top+1;}

   int push(int item) {
       if (isFull()) throw new StackOverflowError("Stack is full.");
       implArr[++top] = item;
       return item;
   }

   int pop() {
       if (isEmpty()) throw new IllegalStateException("Stack is Empty.");
       top--;
       return implArr[top+1];
   }

   boolean isEmpty() { return top == -1; }

   boolean isFull() { return top == (implArr.length-1); }

   int peek() {
       if (isEmpty()) throw new IllegalStateException("Stack is Empty.");
       return implArr[top+1];
   }
```

```java
    void display() {
        if (isEmpty()) throw new IllegalStateException("Stack is Empty.");
        System.out.print("                      +");
        for (int i=top; i>=0; i--) {
            System.out.printf("-%s-+", "-".repeat(Integer.toString(implArr[i]).length()));
        }
        System.out.println();
        System.out.print("Stack Display (LIFO): |");
        for (int i=top; i>=0; i--) {
            System.out.printf(" %d |", implArr[i]);
        }
        System.out.println();
        System.out.print("                      +");
        for (int i=top; i>=0; i--) {
            System.out.printf("-%s-+", "-".repeat(Integer.toString(implArr[i]).length()));
        }
        System.out.println();
    }
}
```

Test:

```java
public class Test {
    public static void main(String[] args) throws NoSuchMethodException {
        int l1 = 10;
        int n = 5;
        System.out.println("\n************ Stack Test ************");
        Stack s = new Stack(l1);
        for (int i = 1; i <= n; i++) {
            System.out.printf("Pushed item %s to the stack s.\n", s.push(i));
        }
        s.display();
        System.out.printf("Popped item %d from the stack.\n", s.pop());
        System.out.printf("Top of Stack: %d.\n", s.getTop());
        System.out.printf("Top element in stack: %d.\n", s.peek());
        s.display();
    }
}
```

Output:

```
Run    Test
"C:\Program Files\Java\jdk-24.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.1\lib\idea_rt.jar=60836" -Dfile.encoding=UTF-8 -Dsun.s
Stack Test

Pushed item 1 to the stack s.
Pushed item 2 to the stack s.
Pushed item 3 to the stack s.
Pushed item 4 to the stack s.
Pushed item 5 to the stack s.
                      +---+---+---+---+---+
Stack Display (LIFO): | 5 | 4 | 3 | 2 | 1 |
                      +---+---+---+---+---+
Popped item 5 from the stack.
Top of Stack: 4.
Top element in stack: 5.
                      +---+---+---+---+
Stack Display (LIFO): | 4 | 3 | 2 | 1 |
                      +---+---+---+---+
```

2. Queue Implementation

Code:

```java
import java.util.NoSuchElementException;

public class Queue {
   int[] implArr;
   int length;
   int front = 0;
   int rear = 0;

   public Queue(int length) {
       this.length = length;
       implArr = new int[length];
   }

   public boolean isEmpty() {
       return front == rear;
   }

   public boolean isFull() {
       return front == (rear+1)%length;
   }

   int enqueue(int item) {
       if (isFull()) throw new IllegalStateException("Queue is full.");
       else {
           implArr[rear] = item;
           rear = (rear+1)%length;
       }
       return item;
   }
```

```java
    int dequeue() {
        if (isEmpty()) throw new NoSuchElementException("Queue is empty.");
        else {
            int[] newArr = new int[length];
            for (int i = 0; i< implArr.length; i++) {
                int itemInQueueArr = implArr[i];
                if (i!=front) newArr[i] = itemInQueueArr;
            }
            implArr = newArr;
            front = (front+1)%(length);
        }
        return implArr[front];
    }

    void display() {
        if (isEmpty()) throw new IllegalStateException("Queue is Empty.");
        System.out.print("                      +");
        for (int i=rear-1; i>=0; i--) {
            System.out.printf("-%s-+", "-".repeat(Integer.toString(implArr[i]).length()));
        }
        System.out.println();
        System.out.print("Queue Display (FIFO): |");
        for (int i=rear-1; i>=0; i--) {
            System.out.printf(" %d |", implArr[i]);
        }
        System.out.println();
        System.out.print("                      +");
        for (int i=rear-1; i>=0; i--) {
            System.out.printf("-%s-+", "-".repeat(Integer.toString(implArr[i]).length()));
        }
        System.out.println();
    }
}
```

Test:

```java
public class Test {
    public static void main(String[] args) throws NoSuchMethodException {
        int l1 = 8;
        int n = 5;

        System.out.println("Queue Test\n");
        Queue q = new Queue(l1);
        for (int i = 1; i <= n; i++) {
            System.out.printf("Added item %s to queue s.\n", q.enqueue(i));
        }
        q.display();
        System.out.printf("Removed item %d from queue.\n", q.dequeue());
```

```
        System.out.printf("Removed item %d from queue.\n", q.dequeue());
        for (int i = 6; i <= 8; i++) {
            System.out.printf("Added item %s to queue s.\n", q.enqueue(i));
        }
        q.display();
    }
}
```

Output:

```
Run    Test ×
"C:\Program Files\Java\jdk-24.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.1\lib\idea_rt.jar=60785" -Dfile.encoding=UTF-8 -Dsun.s
Queue Test

Added item 1 to queue s.
Added item 2 to queue s.
Added item 3 to queue s.
Added item 4 to queue s.
Added item 5 to queue s.

                      +---+---+---+---+---+
Queue Display (FIFO): | 5 | 4 | 3 | 2 | 1 |
                      +---+---+---+---+---+
Removed item 2 from queue.
Removed item 3 from queue.
Added item 6 to queue s.
Added item 7 to queue s.
Added item 8 to queue s.

                      +---+---+---+---+---+---+---+---+
Queue Display (FIFO): | 8 | 7 | 6 | 5 | 4 | 3 | 0 | 0 |
                      +---+---+---+---+---+---+---+---+

Process finished with exit code 0
```

3. Array Implementation

Code:

```
public class Array {
    private int[] implArr;
    private int index = -1;

    public Array(int length) {
        this.implArr = new int[length];
    }

    int insert(int item) {
        if (isFull()) throw new ArrayIndexOutOfBoundsException("Array is Full.");
        implArr[++index] = item;
        return item;
    }

    int delete(int pos) {
```

```java
        if (isEmpty()) throw new IllegalStateException("Array is Empty.");
        int[] swapArr = new int[implArr.length];
        for (int i = 0; i < pos; i++) {
            swapArr[i] = implArr[i];
        }
        for (int i = pos; i < implArr.length-1; i++) {
            swapArr[i] = implArr[i+1];
        }
        implArr = swapArr;
        index--;
        return pos;
    }

    boolean isEmpty() { return index == -1; }

    boolean isFull() { return index == (implArr.length-1); }

    void display() {
        System.out.print("Array Display: [ ");
        for (int i = 0; i <= index; i++) {
            System.out.printf("%d%s", implArr[i], i!=(index)?", ":" ]\n");
        }
    }
}
```

Test:

```java
public class Test {
    public static void main(String[] args) throws NoSuchMethodException {
        int l1 = 10;
        int n = 5;

        System.out.println("Array Test\n");
        Array a = new Array(l1);
        for (int i = 1; i <= n; i++) {
            System.out.printf("Inserted item %s to array a.\n", a.insert(i));
        }
        a.display();
        System.out.printf("Deleted element at index %d.\n", a.delete(3));
        System.out.printf("Deleted element at index %d.\n", a.delete(2));
        a.display();
    }
}
```

Output:

```
Run    Test
"C:\Program Files\Java\jdk-24.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.1\lib\idea_rt.jar=60832" -Dfile.encoding=UTF-8 -Dsun.s
Array Test

Inserted item 1 to array a.
Inserted item 2 to array a.
Inserted item 3 to array a.
Inserted item 4 to array a.
Inserted item 5 to array a.
Array Display: [ 1, 2, 3, 4, 5 ]
Deleted element at index 3.
Deleted element at index 2.
Array Display: [ 1, 2, 5 ]
```

4. Single Linked List Implementation

Code:

```java
import java.util.ArrayList;

public class SinglyLinkedList {
   ArrayList<Node> nodes = new ArrayList<>();
   int pointer = 0;

   Node head() {
       return isEmpty() ? null : nodes.getFirst();
   }

   Node tail() {
       return isEmpty() ? null : nodes.getLast();
   }

   boolean isEmpty() { return pointer == 0; }

   int add(int data) {
       Node node = new Node(this, data);
       if (pointer >= 1) {
           nodes.add(node);
           nodes.get(pointer-1).next = node;
       }
       else nodes.add(node);
       pointer++;
       return data;
   }

   int remove(int item) throws NoSuchMethodException {
       for (int i = 0; i < pointer; i++) {
           if (nodes.get(i).data == item) {
               if (head().data == item) {
                   nodes.remove(i);
```

```
                    pointer--;
                    return item;
                }
                if (tail().data == item) {
                    nodes.get(i-1).next = null;
                    nodes.remove(i);
                    pointer--;
                    return item;
                }
                nodes.get(i-1).next = nodes.get(i+1);
                nodes.remove(i);
                pointer--;
                return item;
            }
        }
        pointer--;
        return item;
    }

    int count() {
        return pointer;
    }

    void display() {
        System.out.print("\n                    ");
        for (Node n : nodes) {
            System.out.printf("+--%s--+-%s-+%s", "-".repeat(n.toString().length()),
n.next!=null?"-".repeat(n.next.toString().length()):"----", (n.next)!=null ? "    " : "");
        }
        System.out.println();
        System.out.print("Singly Linked List: ");
        for (Node n : nodes) {
            System.out.printf("|  %s  | %s |%s", n, n.next, (n.next)!=null ? " -> " : "");
        }
        System.out.println();
        System.out.print("                    ");
        for (Node n : nodes) {
            System.out.printf("+--%s--+-%s-+%s", "-".repeat(n.toString().length()),
n.next!=null?"-".repeat(n.next.toString().length()):"----", (n.next)!=null ? "    " : "");
        }
        System.out.println();
        System.out.println();
    }
}

class Node {
    SinglyLinkedList parent;
    Node next;
```

```java
    int data;

    public Node(SinglyLinkedList parent, int data) {
        this.parent = parent;
        this.next = null;
        this.data = data;
    }

    public String toString() { return String.format("%d", this.data); }
}
```

Test:

```java
public class Test {
    public static void main(String[] args) throws NoSuchMethodException {
        int l1 = 10;
        int n = 5;

        System.out.println("Singly Liked List Test\n");
        SinglyLinkedList sll = new SinglyLinkedList();
        for (int i = 1; i <= n; i++) {
            System.out.printf("Added item %s to Linked list sll.\n", sll.add(i));
        }
        sll.display();
        System.out.printf("Removed item %d from Singly Linked List ssl.\n", sll.remove(1));
        System.out.printf("Removed item %d from Singly Linked List ssl.\n", sll.remove(3));
        System.out.printf("Removed item %d from Singly Linked List ssl.\n", sll.remove(5));
        sll.display();
    }
}
```

Output:

```
Run    Test
"C:\Program Files\Java\jdk-24.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.1\lib\idea_rt.jar=60825" -Dfile.encoding=UTF-8 -Dsun.s
Singly Liked List Test

Added item 1 to Linked list sll.
Added item 2 to Linked list sll.
Added item 3 to Linked list sll.
Added item 4 to Linked list sll.
Added item 5 to Linked list sll.

                    +-----+---+    +-----+---+    +-----+---+    +-----+---+    +-----+------+
Singly Linked List: |  1  | 2 | -> |  2  | 3 | -> |  3  | 4 | -> |  4  | 5 | -> |  5  | null |
                    +-----+---+    +-----+---+    +-----+---+    +-----+---+    +-----+------+

Removed item 1 from Singly Linked List ssl.
Removed item 3 from Singly Linked List ssl.
Removed item 5 from Singly Linked List ssl.

                    +-----+---+    +-----+------+
Singly Linked List: |  2  | 4 | -> |  4  | null |
                    +-----+---+    +-----+------+
```

5. Double Linked List Implementation

Code:

```java
import java.util.ArrayList;

public class DoubleLinkedList extends SinglyLinkedList {
   ArrayList<DNode> nodes = new ArrayList<>();

   DNode head() {
       return isEmpty() ? null : nodes.getFirst();
   }

   DNode tail() {
       return isEmpty() ? null : nodes.getLast();
   }

   int add(int data) {
       DNode node = new DNode(this, data);
       if (pointer >= 1) {
           nodes.get(pointer-1).next = node;
           node.prev = nodes.get(pointer-1);
       }
       nodes.add(node);
       pointer++;
       return data;
   }

   int remove(int item) {
       for (int i = 0; i < pointer; i++) {
           if (nodes.get(i).data == item) {
               // first item
               if (head().data == item) {
                   nodes.remove(i);
                   head().prev = null;
                   pointer--;
                   return item;
               }

               // last item
               if (tail().data == item) {
                   nodes.get(i-1).next = null;
                   nodes.remove(i);
                   pointer--;
                   return item;
               }
```

```
                // middle item
                nodes.get(i-1).next = nodes.get(i+1);
                nodes.get(i+1).prev = nodes.get(i-1);
                nodes.remove(i);
                pointer--;
                return item;
            }
        }
        pointer--;
        return item;
    }

    void displayEngine(String displayName) {
        System.out.printf("\n%s", " ".repeat(displayName.length()+2));
        for (DNode n : nodes) {
            System.out.printf("+-%s-+--%s--+-%s-+%s", n.prev!=null?"-".repeat(n.prev.toString().length()):"----",
"-".repeat(n.toString().length()), n.next!=null?"-".repeat(n.next.toString().length()):"----", (n.next)!=null ?
"     " : "");
        }
        System.out.println();
        System.out.printf("%s: ", displayName);
        for (DNode n : nodes) {
            System.out.printf("| %s |  %s  | %s |%s", n.prev, n, n.next, (n.next)!=null ? " <-> " : "");
        }
        System.out.println();
        System.out.printf("%s", " ".repeat(displayName.length()+2));
        for (DNode n : nodes) {
            System.out.printf("+-%s-+--%s--+-%s-+%s", n.prev!=null?"-".repeat(n.prev.toString().length()):"----",
"-".repeat(n.toString().length()), n.next!=null?"-".repeat(n.next.toString().length()):"----", (n.next)!=null ?
"     " : "");
        }
        System.out.println();
        System.out.println();
    }

    void display() {
        displayEngine("Double Linked List");
    }
}

class DNode extends Node {
    DNode next;
    DNode prev;

    public DNode(SinglyLinkedList parent, int data) {
        super(parent, data);
        this.prev = null;
    }
```

```
}
```

Test:

```java
public class Test {
   public static void main(String[] args) throws NoSuchMethodException {
       int l1 = 10;
       int n = 5;

       System.out.println("Double Liked List Test\n");
       DoubleLinkedList dll = new DoubleLinkedList();
       for (int i = 1; i <= n; i++) {
           System.out.printf("Added item %s to Linked list dll.\n", dll.add(i));
       }
       dll.display();
       System.out.printf("Removed item %d from Double Linked List dll.\n", dll.remove(1));
       System.out.printf("Removed item %d from Double Linked List dll.\n", dll.remove(3));
       System.out.printf("Removed item %d from Double Linked List dll.\n", dll.remove(5));
       dll.display();
   }
}
```

Output:

```
Run    Test
"C:\Program Files\Java\jdk-24.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.1\lib\idea_rt.jar=60818" -Dfile.encoding=UTF-8 -Dsun.s
Double Liked List Test

Added item 1 to Linked list dll.
Added item 2 to Linked list dll.
Added item 3 to Linked list dll.
Added item 4 to Linked list dll.
Added item 5 to Linked list dll.

                   +------+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+------+
Double Linked List: | null |  1  | 2 | <-> | 1 |  2  | 3 | <-> | 2 |  3  | 4 | <-> | 3 |  4  | 5 | <-> | 4 |  5  | null |
                   +------+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+------+

Removed item 1 from Double Linked List dll.
Removed item 3 from Double Linked List dll.
Removed item 5 from Double Linked List dll.
Dll count: 2.

                   +------+-----+---+     +---+-----+------+
Double Linked List: | null |  2  | 4 | <-> | 2 |  4  | null |
                   +------+-----+---+     +---+-----+------+
```

6. Circular Double Linked List Implementation

Code:

```java
public class CircularDoubleLinkedList extends DoubleLinkedList{
    int add(int data) {
        DNode node = new DNode(this, data);
        if (!isEmpty()){
            nodes.get(pointer-1).next = node;
            node.prev = nodes.get(pointer-1);
            head().prev=node;
        }
        else {
            node.prev = node.next = node;
        }
        node.next = head() != null ? head() : node;
        nodes.add(node);
        pointer++;
        return data;
    }

    int remove(int item) {
        for (int i=0; i < pointer; i++) {
            if (nodes.get(i).data == item) {
                if (head().data == item) {
                    nodes.remove(i);
                    tail().next = head();
                    head().prev = tail();
                    pointer--;
                    return item;
                }
                if (tail().data == item) {
                    nodes.remove(i);
                    nodes.get(i-1).next = head();
                    head().prev=tail();
                    pointer--;
                    return item;
                }
                nodes.get(i-1).next = nodes.get(i+1);
                nodes.get(i+1).prev = nodes.get(i-1);
                nodes.remove(i);
                pointer--;
                return item;
            }
        }
        pointer--;
        return item;
    }

    @Override
    void display() {
        displayEngine("Circular Double Linked List");
```

```
    }
}
```

Test:

```java
public class Test {
    public static void main(String[] args) throws NoSuchMethodException {
        int l1 = 10;
        int n = 5;
        System.out.println("Circular Double Liked List Test\n");
        CircularDoubleLinkedList cdll = new CircularDoubleLinkedList();
        for (int i = 1; i <= n; i++) {
            System.out.printf("Added item %s to Linked list cdll.\n", cdll.add(i));
        }
        cdll.display();
        System.out.printf("Removed item %d from Circular Double Linked List cdll.\n", cdll.remove(1));
        System.out.printf("Removed item %d from Circular Double Linked List cdll.\n", cdll.remove(3));
        System.out.printf("Removed item %d from Circular Double Linked List cdll.\n", cdll.remove(5));
        cdll.display();
    }
}
```

Output:

```
Run    Test
"C:\Program Files\Java\jdk-24.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.1\lib\idea_rt.jar=60849" -Dfile.encoding=UTF-8 -Dsun.s
Circular Double Liked List Test

Added item 1 to Linked list cdll.
Added item 2 to Linked list cdll.
Added item 3 to Linked list cdll.
Added item 4 to Linked list cdll.
Added item 5 to Linked list cdll.

                           +---+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+---+
Circular Double Linked List: | 5 |  1  | 2 | <-> | 1 |  2  | 3 | <-> | 2 |  3  | 4 | <-> | 3 |  4  | 5 | <-> | 4 |  5  | 1 | <->
                           +---+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+---+     +---+-----+---+

Removed item 1 from Circular Double Linked List cdll.
Removed item 3 from Circular Double Linked List cdll.
Removed item 5 from Circular Double Linked List cdll.

                           +---+-----+---+     +---+-----+---+
Circular Double Linked List: | 4 |  2  | 4 | <-> | 2 |  4  | 2 | <->
                           +---+-----+---+     +---+-----+---+
```

NAME: IGWENAGU CHINWEIKE OBIORA

REGISTRATION NUMBER: 2023/257473

DEPARTMENT: COMPUTER SCIENCE

TITLE: COS 232 ASSIGNMENT

Object: Stack
Attributes:
☐ implArr ☐ length ☐ Top
Behaviours:
☐ push ☐ pop ☐ peek ☐ isEmpty ☐ isFull ☐ display

Object: Array
Attributes:
☐ implArray ☐ index
Behaviours:
☐ insert ☐ delete ☐ isEmpty ☐ isFull

Object: SinglyLinkedList
Attributes:
☐ nodes ☐ pointer
Behaviours:
☐ head ☐ tail ☐ isEmpty ☐ add ☐ remove ☐ count ☐ display

Object: DoubleLinkedList
Attributes:
☐ nodes ☐ pointer
Behaviours:
☐ head ☐ tail ☐ add ☐ remove ☐ count ☐ displayEngine ☐ display

Object: CircularDoubleLinkedList
Attributes:
☐ nodes ☐ pointer
Behaviours
☐ head ☐ tail ☐ add ☐ remove ☐ count ☐ displayEngine ☐ display

Object: Queue
Attributes:
☐ implArr ☐ length ☐ front ☐ rear ☐ displayrear
Behaviours
☐ isEmpty ☐ isFull ☐ enqueue ☐ dequeue ☐ display

Object: Node
Attributes:
☐ parent ☐ next ☐ data

Object: DNode
Attributes:
☐ parent ☐ prev ☐ next ☐ data