



**COMSATS University**  
**Islamabad**  
**Abbottabad Campus**

**SUBMITTED BY:**                      **ZULQARNAIN HAIDER**

**REG #:**                                      **SP24-BSE-179**

**SUBJECT:**                                **Data Structure Lab 3**

**SUBMITTED TO:**                      **Nauman khan**

**DATE:**                                      **17-09-2025**

---

### **Lab Task 03**

#### **Data structure concept (Array Implementation)**

##### **Q1: Array data structure:**

###### **Instruction**

Write a Java program to implement a menu-driven system for the following operations on an integer array and also design algorithm:

1. Traversal: • Display all elements in the array taken from the user.
2. Insertion: Insert an element: • At Beginning • At Given Position (take position from user) • At End
3. Deletion: Delete an element: • From Beginning • From Given Position (take position from user) • From End
4. Searching: • Linear Search (take value of key from user)
5. Exit

```

import java.util.Scanner;

public class ArrayOperationMenu {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int[] arr = new int[100];
        int n;

        System.out.print("Enter number of elements: ");
        n = sc.nextInt();

        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        int choice;
        do {
            System.out.println("\n--- ARRAY MENU ---");
            System.out.println("1. Traversal");
            System.out.println("2. Insert Element");
            System.out.println("3. Delete Element");
            System.out.println("4. Search Element");
            System.out.println("5. Exit");
            System.out.print("Choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Array: ");
                    for (int i = 0; i < n; i++) System.out.print(arr[i] + " ");
                    System.out.println();
                    break;

                case 2:
                    System.out.print("Enter element: ");
                    int ele = sc.nextInt();
                    System.out.print("Enter position (1 to " + (n + 1) + "): ");
                    int pos = sc.nextInt();
                    if (pos >= 1 && pos <= n + 1) {
                        for (int i = n; i >= pos; i--) arr[i] = arr[i - 1];
                        arr[pos - 1] = ele;
                        n++;
                    } else System.out.println("Invalid position!");
                    break;

                case 3:
                    System.out.print("Enter position (1 to " + n + "): ");
                    int dpos = sc.nextInt();
                    if (dpos >= 1 && dpos <= n) {
                        for (int i = dpos - 1; i < n - 1; i++) arr[i] = arr[i + 1];
                        n--;
                    } else System.out.println("Invalid position!");
                    break;

                case 4:
                    System.out.print("Enter element to search: ");
                    int key = sc.nextInt();
                    int i;
                    for (i = 0; i < n; i++) {
                        if (arr[i] == key) {
                            System.out.println("Found at position " + (i + 1));
                            break;
                        }
                    }
                    if (i == n) System.out.println("Not found!");
                    break;

                case 5:
                    System.out.println("Exit...");
                    break;

                default:
                    System.out.println("Invalid choice!");
            }
        } while (choice != 5);

        sc.close();
    }
}

```

## Output:

```
C:\Users\DELL\jdk\openjdk-24.0.1\bin\java.exe "-javaagent:D:\IntelliJ\IntelliJ IDEA Community Edition 2025.1.4.1\lib\idea_rt.jar=59893" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "C:\Users\DELL\IdeaProjects\ARRAY OPERATIONS MENU\out\production\ARRAY OPERATIONS MENU" ArrayOperationMenu
Enter number of elements: 4
Enter 4 elements:
1
2
3
4

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 3
Enter position (1 to 4): 2

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 4
Enter element to search: 2
Not found!

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 4
Enter element to search: 1
Found at position 1

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 4
Enter element to search: 3
Found at position 2

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 4
Enter element to search: 4
Found at position 3

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 4
Enter element to search: 5
Not found!

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 1
Array: 1 3 4

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 2
Enter element: 44
Enter position (1 to 4): 1

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 1
Array: 44 1 3 4

--- ARRAY MENU ---
1. Traversal
2. Insert Element
3. Delete Element
4. Search Element
5. Exit
Choice: 5
Exit...

Process finished with exit code 0
```

## **Algorithm: Array Operation Menu**

- **Step 1:** Start
- **Step 2:** Create an array `arr[100]`
- **Step 3:** Input `n` (number of elements)
- **Step 4:** Enter `n` elements into the array
- **Step 5:** Repeat until the user chooses Exit:
  - **Step 5.1:** Display the menu
  - **Step 5.2:** Input user's choice
  - **Step 5.3:** If choice = 1 → Display all elements (Traversal)
  - **Step 5.4:** If choice = 2 → Insert element at given position
    - Shift elements to the right
    - Place new element
    - Increase `n` by 1
  - **Step 5.5:** If choice = 3 → Delete element from given position
    - Shift elements to the left
    - Decrease `n` by 1
  - **Step 5.6:** If choice = 4 → Search element in the array (Linear Search)
    - If found → show position
    - Else → show "Not Found"
  - **Step 5.7:** If choice = 5 → Exit program
- **Step 6:** Stop