

**AIM:** Design develop and implement a menu driven program for the following Array operations.

- Creating an Array of N integer Elements
- Display of Array Elements with suitable Readings.
- Inserting an Element at a given valid position (Pos).
- Deleting an Element at a given valid position.
- Exit.

Support the program with functions for each of the above functions.

### ALGORITHM.

- Start
- Initialize Variables:
  - array as an array of int (initially null).
  - Size as an int (initially 0).
- Define a fn CreateArray().
- (a) Display "Enter size of array"
- (b) Read Size from user
- (c) Create a new array with size elements and store it in array variable.
- (d) display "Enter value of array"
- (e) repeat for  $i = 0$  to size - 1.
- (f) Read element from user and store in array [i]
- (g) Display "Array of size" + size + " created".

Teacher's Signature : \_\_\_\_\_

- (4) Define a function `displayArray();`
- If array is null, display "Array is empty"
  - Else, display "Array Elements with Headings"
    - Display "Index It. Value"
    - Repeat for  $i=0$  to  $\text{size}-1$ ;
    - $\text{display} i + \text{"It}\backslash\text{t"} + \text{array}[i]$ .
- (5) Define a function `insertElement();`
- If array is null, display "Array is Empty"
  - Else:
    - Display "Enter position to insert element"
    - Read pos from the user.
    - If  $\text{pos} < 0$  ||  $\text{pos} > \text{size}$ , display "invalid position!"
    - else:
      - Display "Element to insert"
      - Read element from user
      - Create a new array 'newArray' of  $\text{size}+1$ .
      - Repeat for  $i=0$  to  $\text{pos}-1$ ;
        - Copy array[i] to newArray[i]
      - Set  $\text{newArray}[\text{pos}] = \text{element}$
      - Repeat for  $i=\text{pos}+1$  to  $\text{size}$ ;
        - Copy array[i-1] to newArray[i]
      - Set array = newArray.
      - Increment size by 1
      - Display "Element inserted at pos" + pos"!"

Teacher's Signature : \_\_\_\_\_

(6) Define fn deleteElement():  
follow similar steps as in 5.(insertElement),  
change else:

1. Create a new array "newArray" of size-1
2. Repeat for  $i=0$  to  $pos-1$ .
  - Copy array[i] to newArray[i]
3. Repeat for  $i=pos$  to  $size-1$ .
  - Copy array[i+1] to newArray[i]
4. Set array = newArray
5. Decrement size by 1.
6. Display "Element from position pos".

(7) Display menu options and perform the following until exist.

- (a). Display menu options.
- (b). "a": Display array "
- "b. Create array"
- "c Insert element"
- "d delete Element"
- "e Exit".

(b) Read the user's choice.

- (c) If choice is 'a' , Call CreateArray().
- (d) If choice is 'b' , Call display Array().
- (e) If choice is 'c' Call insertElement().
- (f) If choice is 'd' Call delete Element().

(g) End.

CODE :

```

import java.util.Scanner;

public class ArrayOperationsMenu {
    private static Scanner scanner = new Scanner(System.in);
    private static int[] array;
    private static int size = 0;

    public static void main(String[] args) {
        boolean exit = false;

        while (!exit) {
            System.out.println("Menu");
            System.out.println("a. Create array");
            System.out.println("b. Display array");
            System.out.println("c. Insert Element");
            System.out.println("d. Delete Element");
            System.out.println("e. Exit");
            System.out.println("Enter your choice");

            char choice = scanner.next().charAt(0);

            switch(choice) {
                case 'a': CreateArray();
                break;
                case 'b': displayArray();
                break;
                case 'c': insertElement(); break;
            }
        }
    }

    private static void CreateArray() {
        array = new int[size];
        for (int i = 0; i < size; i++) {
            array[i] = scanner.nextInt();
        }
    }

    private static void displayArray() {
        for (int i = 0; i < size; i++) {
            System.out.print(array[i] + " ");
        }
        System.out.println();
    }

    private static void insertElement() {
        int index;
        System.out.println("Enter index to insert element");
        index = scanner.nextInt();
        if (index > size) {
            System.out.println("Index out of bound");
            return;
        }
        System.out.println("Enter element to insert");
        int element = scanner.nextInt();
        array[index] = element;
        size++;
    }

    private static void deleteElement() {
        int index;
        System.out.println("Enter index to delete element");
        index = scanner.nextInt();
        if (index > size) {
            System.out.println("Index out of bound");
            return;
        }
        size--;
        for (int i = index; i < size; i++) {
            array[i] = array[i + 1];
        }
    }
}

```

Teacher's Signature : \_\_\_\_\_

```

Case 'd': deleteElement(); break;
Case 'e': exit = true;
System.out.println("Exiting program");
break;

```

default:

```

System.out.println("Invalid choice");
break;

```

3.

```
Scanner.close();
```

2.

```

private static void createArray() {
    System.out.println("Enter size of array");
    Size = scanner.nextInt();
    array = new int[Size];
}

```

```
for(int i=0; i < Size; i++) {
```

```

        System.out.println("Element at position " + i + ":");
        array[i] = scanner.nextInt();
}

```

3.

```
System.out.println("Array of size " + Size +
    " created");
```

4.

```
private static void deleteArray() {
    if(Size == 0) {
```

```

        System.out.println("Array empty");
    }
}
```

Teacher's Signature :

```
return; }  
System.out.println("Enter position");  
int pos = scanner.nextInt();  
  
if (pos < 0 || pos >= size)  
{  
    System.out.println("Invalid position");  
    return;  
}  
  
int[] newArray = new int[size - 1];  
  
for (int i = 0; i < pos; i++) {  
    newArray[i] = array[i]; }  
  
for (int i = pos; i < size - 1; i++) {  
    newArray[i] = array[i + 1]; }  
  
array = newArray;  
size--;  
System.out.println("Element deleted from  
position " + pos + ".");  
  
}  
  
private static void displayArray()  
{  
    if (size == 0) {  
        System.out.println("Array is empty");  
        return;  
    }
```

Teacher's Signature : \_\_\_\_\_

```

System.out.println("Array elements");
System.out.println("Individual value at " + i);
for (int i = 0; i < size; i++) {
    System.out.print(i + " " + array[i]);
}
}

```

```

private static void insertElement() {
    if (size == 0) {
        System.out.println("Empty array");
        return;
    }
}

```

```

System.out.print("Enter position to insert element");
int pos = scanner.nextInt();
if (pos < 0 || pos > size) {
    System.out.println("Invalid position");
    return 0;
}

```

```

System.out.println("Enter Element to insert");
int element = scanner.nextInt();

```

```

int[] newArray = new int[size + 1];
for (int i = 0; i < pos; i++) {
    newArray[i] = array[i];
}
newArray[pos] = element;

```

```
for (int i = pos+1; i < size + 1; i++) {  
    newArray[i] = array[i-1];  
}
```

array = newArray;  
size += 1;  
System.out.println("Element at  
position " + pos + " is " +  
array);

3.

