ABES Engineering College, Ghaziabad.

Affiliated to Dr. A.P.J Abdul Kalam Technical University, Lucknow.

Department of CSE-Data Science

Title	**Lecture Notes**
Subject	**Programming for Problem Solving**
Topics	**Array**
Lecture Date	**30 May 2024**
Faculty Name	**Mr. Dilip Kr. Bharti**
Section	**First Year Section-B**

Syllabus:

Arrays: Array notation and representation, manipulating array elements, using multi dimensional arrays.

Basic Searching and Sorting Algorithms: Searching &Basic Sorting Algorithms (Bubble, Insertion and Selection).

What is an Array?

Definition:

- **Collection of Elements:**
  - An array is a collection of elements, each of which is of the same data type.
- **Contiguous Memory Locations:**
  - Elements are stored in contiguous memory locations.
- **Fixed Size:**
  - The size of an array is fixed at the time of declaration and cannot be changed.
- **Indexing:**
  - Each element in the array is accessed using an index, with the first element having an index of 0.
- **Homogeneous Data Type:**
  - All elements in the array must be of the same type (e.g., all integers, all floats, etc.).
- **Direct Access:**

- Arrays allow direct access to any element using its index, making data retrieval fast and efficient.

Purpose:

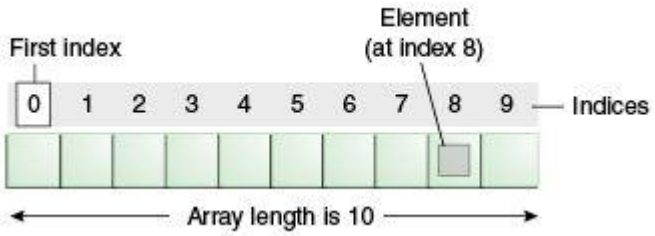
Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

Array Type:

- ✓ **One-Dimensional Arrays:** Linear lists of elements.
- ✓ **Two-Dimensional Arrays:** Matrix-like structures.
- ✓ **Multi-Dimensional Arrays:** Higher-dimensional structures.
- ✓ **Character Arrays:** Arrays of characters forming strings.
- ✓ **Array of Pointers:** Arrays where each element is a pointer.
- ✓ **Dynamic Arrays:** Arrays with size determined at runtime.

One-Dimensional Arrays

- ✓ **Declaring Arrays:** Define the type and size.
- ✓ **Initializing Arrays:** Provide initial values within curly braces.
- ✓ **Accessing Elements:** Use indices within square brackets.
- ✓ **Calculating Length:** Use sizeof() operator.
- ✓ **Iterating Over Arrays:** Use loops to traverse elements.



Declaring and Initializing Arrays

- **Syntax for Declaration:**

  Data_type array_Name[arraySize];

- **Initialization During Declaration:**

```
int numbers[5] = {1, 2, 3, 4, 5};
```

- **Default Initialization:**

```
int numbers[5] = {0}; // All elements set to 0
int numbers[5]={1,2}; //Rest all element set to 0
int numbers[]={1,3,4,5}; //Set Array size automatically.
Int numbers[5]={1,2,3,4,5,6,7,8}// Rest elements is not accessible.
```

Accessing Array Elements:

- **Indexing:** Array elements are accessed using their index.
- **Index Starts at 0:** The first element is at index 0.

Example:

```
int firstNumber = numbers[0]; // Accesses the first element
numbers[2] = 10; // Sets the third element to 10
```

Array Traversal

- **Using Loops:** Commonly, arrays are traversed using loops.

Example:

```
for(i = 0; i < 5; i++)
{
   printf("%d ", numbers[i]);
}
```

Example:

Write a program to input elements and print elements from user?

Calculating the Length of an Array Using sizeof() Operator :

The sizeof() operator in C can be used to determine the size of an array in bytes. By dividing the total size of the array by the size of a single element, we can find the number of elements in the array.

Steps to Calculate the Length of an Array:

1. **Get the total size of the array:**
   - Use sizeof(arrayName) to get the total size in bytes.
2. **Get the size of a single element:**
   - Use sizeof(arrayName[0]) to get the size of one element in bytes.
3. **Calculate the number of elements:**
   - Divide the total size by the size of one element.

Formula:

```
int length = sizeof(arrayName) / sizeof(arrayName[0]);
```

```
#include <stdio.h>

int main() {
   int numbers[] = {1, 2, 3, 4, 5};

   // Calculate the length of the array
   int length = sizeof(numbers) / sizeof(numbers[0]);

   // Print the length
   printf("Length of the array: %d\n", length);

   return 0;
}
```

Explanation:

- sizeof(numbers) returns the total size of the array in bytes.
- sizeof(numbers[0]) returns the size of the first element in bytes.
- Dividing these two values gives the number of elements in the array.

Output:

```
Length of the array: 5
```

Advantages and Limitations

- **Advantages:**
  - Easy to use.
  - Efficient memory usage.
  - Fast access to elements.
  - Sorting and Searching is fast.
- **Limitations:**
  - Fixed size.
  - Contiguous memory allocation.
  - Insertion and deletion operations are complicated (Using by Shifting).

Practice Program:

1. Write a program in C to store elements in an array and print them.

Input 10 elements in the array:
element - 0 : 1
element - 1 : 1
element - 2 : 2
Elements in array are: 1 1 2 3 4 5 6 7 8 9

2. Write a program in C to find the sum of all elements of the array.

3. Write a program in C to find the maximum and minimum elements in an array.

4. Write a program in C to separate odd and even integers into separate arrays.

5. Write a program in C to find the second largest element in an array.

Two-Dimensional Array:

```
#include <stdio.h>

int main() {
   int matrix[3][4] = {
      {1, 2, 3, 4},
      {5, 6, 7, 8},
```

```
    {9, 10, 11, 12}
  };

  // Calculate the number of rows
  int rows = sizeof(matrix) / sizeof(matrix[0]);

  // Calculate the number of columns
  int columns = sizeof(matrix[0]) / sizeof(matrix[0][0]);

  // Print the number of rows and columns
  printf("Number of rows: %d\n", rows);
  printf("Number of columns: %d\n", columns);

  return 0;
}
```

Explanation:

- sizeof(matrix) returns the total size of the 2D array in bytes.
- sizeof(matrix[0]) returns the size of one row.
- sizeof(matrix[0][0]) returns the size of one element.

Output:

```
Number of rows: 3
Number of columns: 4
```