Module: 2

Day5:
Trainer: Asif Nafees

Array

In C programming, an array is a collection of elements of the same data type stored in contiguous memory locations. Arrays are used to store multiple values in a single variable, which can be accessed using an index. Here's an in-depth explanation of arrays in C:

1. Declaration of Arrays

Arrays must be declared before use. The declaration syntax is:
data_type array_name[array_size];

data_type: The type of elements the array will hold (e.g., int, float, char).
array_name: The name of the array.
array_size: The number of elements the array will hold (this must be a constant expression).

Example:
int numbers[5];
This declares an array numbers that can store 5 integers.

2. Initialization of Arrays

Arrays can be initialized at the time of declaration or later. You can initialize arrays by assigning values to them in curly braces {}.

Method 1: At the time of declaration:
int numbers[5] = {1, 2, 3, 4, 5};

Method 2: Automatic size determination: When initializing an array, you can omit the size, and the compiler will automatically determine it based on the number of elements:
int numbers[] = {1, 2, 3, 4, 5};

Method 3: Partial initialization: If you provide fewer values than the specified size, the remaining elements will be initialized to 0 (for numeric types) or '\0' (for char arrays).
int numbers[5] = {1, 2}; // Remaining elements will be initialized to 0

3. Accessing Array Elements

Array elements are accessed using the array index, which starts from 0. The syntax is:
array_name[index]

For example:
int numbers[5] = {1, 2, 3, 4, 5};

```
printf("%d", numbers[0]);  // Outputs 1
printf("%d", numbers[4]);  // Outputs 5
```

4. Modifying Array Elements
You can modify the value of an array element using the index:

```
numbers[0] = 10;  // Change the first element to 10
```

5. Important Points about Arrays in C

Fixed size: Once an array is declared, its size cannot be changed during runtime.
Out-of-bounds access: Accessing an array element outside its declared size (e.g., numbers[10] when numbers is declared with size 5) results in undefined behavior, and it might crash the program.

Advantages:
Arrays allow for efficient memory usage when handling large datasets.
Easy access and modification of elements through indexing.

Limitations:

The size of an array is fixed after its declaration.
Arrays cannot hold different types of data (homogeneous data only).
Insertion and deletion of elements in arrays can be inefficient as they may require shifting of elements.

Example Code: Basic Array Operations

```
#include <stdio.h>
int main() {
    int numbers[5] = {10, 20, 30, 40, 50};

    // Accessing array elements
    printf("First element: %d\n", numbers[0]);
    printf("Last element: %d\n", numbers[4]);

    // Modifying array elements
    numbers[2] = 60;
    printf("Modified third element: %d\n", numbers[2]);

    // Loop through the array
    for (int i = 0; i < 5; i++) {
```

```
        printf("%d ", numbers[i]);
    }

    return 0;
}
```

This program will output:

```
First element: 10
Last element: 50
Modified third element: 60
10 20 60 40 50
```