Session 05 C

Thursday, 9 October 2025 11:02 AM

Array in C

An **array** in C is a collection of elements **of the same data type**, stored in **contiguous memory locations**.
It allows you to store multiple values under a single variable name, accessed using **index numbers**.

Declaration of an Array

```
data_type array_name[size];
```
Example:

```
int numbers[5];  // declares an integer array of size 5
```

Initialization of an Array

You can initialize an array while declaring it:

```
int numbers[5] = {10, 20, 30, 40, 50};
```
You can also let the compiler count the size automatically:

```
int numbers[] = {10, 20, 30, 40, 50};
```

Accessing Array Elements

Array elements are accessed using **index numbers**, starting from **0**.

```
printf("%d", numbers[0]); // prints 10
printf("%d", numbers[4]); // prints 50
```

Example Program

```
#include <stdio.h>
int main() {
   int marks[5] = {90, 85, 75, 88, 92};
   int i;
printf("Student Marks:\n");
   for (i = 0; i < 5; i++) {
      printf("marks[%d] = %d\n", i, marks[i]);
   }
return 0;
}
```
Output:

```
Student Marks:
marks[0] = 90
marks[1] = 85
marks[2] = 75
marks[3] = 88
marks[4] = 92
```

Types of Arrays

1. **One-Dimensional Array** – A single list of elements.
   Example: int arr[10];
2. **Two-Dimensional Array** – Used to store data in rows and columns (like a matrix).
   Example: int matrix[3][3];
3. **Multi-Dimensional Array** – Arrays with more than two dimensions.
   Example: int arr[2][3][4];

Example of 2D Array

```
#include <stdio.h>
int main() {
   int matrix[2][3] = {
      {1, 2, 3},
      {4, 5, 6}
   };
for (int i = 0; i < 2; i++) {
      for (int j = 0; j < 3; j++) {
         printf("%d ", matrix[i][j]);
      }
      printf("\n");
   }
   return 0;
}
```
Output:

```
1 2 3
4 5 6
```

Key Points

- Index starts from **0**.
- Array size must be a **constant** in C.
- Stored in **contiguous memory**.
- Useful for **loops and repetitive data** handling.