**📘 Arrays in C**

Arrays in C are used to store **multiple values of the same data type** using a single variable name. This resource covers:

* Fixed-size arrays
* Character arrays
* Passing arrays to functions
* **1. Fixed-Size Arrays**

**What is a Fixed-Size Array?**

A fixed-size array has a size **defined at compile time** and **cannot be changed** during program execution.

**Syntax:**

data\_type array\_name[size];

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

**Accessing Elements:**

Array indices start at **0**.

numbers[0] = 10;

printf("%d", numbers[0]); // Output: 10

for(int i=0;i<5;i++) numbers[i] = i+1;

for(int i=0;i<5;i++) printf(“%d”,numbers[i]);

printf(“\n”);

**Character Arrays (Strings)**

**What is a Character Array?**

A character array is commonly used to store **strings** in C.

Example:

char name[10] = "Ali";

### Important Notes:

* Strings in C are **null-terminated**. That means the last character is '\0'.
* Always ensure there’s **enough space** for the null character.

**scanf()**

char name[20];

scanf("%s", name); // Reads until a space or newline

**Limitation**: Stops at the first whitespace.

**fgets()**

char name[20];

fgets(name, 20, stdin); // Reads a line, including spaces

**Note**: Keeps the newline character if there's space.

**Passing Arrays to Functions**

**Why?**

To **process** or **modify** array values inside a function.

void printArray(int arr[], int size)

{

for (int i = 0; i < size; i++)

{

printf("%d ", arr[i]);

}

}

### Key Points:

* Arrays are passed by **reference** (not copied).
* The size of the array is typically passed as a **separate argument**.

**📝 Summary**

| **Topic** | **Key Point** |
| --- | --- |
| **Fixed-Size Arrays** | Declared with a fixed number of elements |
| **Character Arrays** | Used for storing text; end with a null character '\0' |
| **Passing to Func** | Passed by reference; size should be passed separately |