

EEE 103 - Computer Programming

L7 - Arrays

Arrays in C

Prepared by: Sk Tahmed Salim Rafid

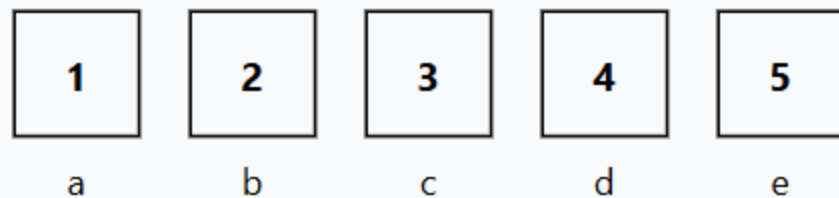
What is an Array?

- ▶ A collection of variables of the same data type stored in contiguous memory.
- ▶ Instead of separate variables like `int a, b, c, d, e;`
- ▶ Use: `int arr[5];`

Visualization: Separate Variables vs Array

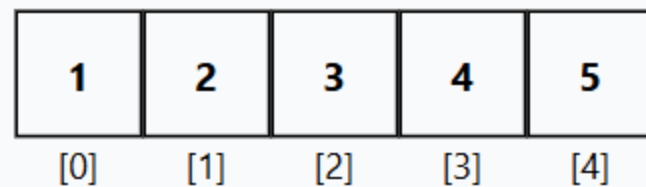
Separate Variables

```
1 int a, b, c, d, e;
```



Array

```
1 int arr[5];
```



Why Do We Use Arrays?

- ▶ Stores multiple values in a single variable.
- ▶ Enables easy data processing with loops.
- ▶ Improves readability and reduces variable clutter.
- ▶ Faster access due to contiguous memory allocation.

Array Declaration

General syntax:

Syntax:

```
data_type array_name[size];
```

Explanation:

- Examples:
- `int marks[10];`
- `float values[5];`

Array Initialization

Ways to initialize arrays:

Syntax:

Explanation:

- (a) Full: `int numbers[5] = {1, 2, 3, 4, 5};`
- (b) Partial (rest=0): `int numbers[5] = {1, 2};`
- (c) Auto size: `int numbers[] = {1, 2, 3, 4, 5};`
- (d) All zero: `int arr[10] = {0};`

Array Initialization Visualization

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

| | | | | |
|----------|----------|----------|----------|----------|
| 1 | 2 | 3 | 4 | 5 |
| [0] | [1] | [2] | [3] | [4] |

Accessing Array Elements

Access using index (starts at 0). For size 5, indices: 0-4.

Syntax:

```
int x = arr[2]; // 3rd element  
arr[0] = 100; // Modify 1st
```

Explanation:

Interactive Array Access

Click buttons to highlight elements in the array.

| | | | | |
|-----------|-----------|-----------|-----------|-----------|
| 10 | 20 | 30 | 40 | 50 |
| [0] | [1] | [2] | [3] | [4] |

Index 0

Index 1

Index 2

Index 3

Index 4

Example: Print All Elements of an Array

```
1 #include <stdio.h>
2 int main() {
3     int arr[5] = {10, 20, 30, 40, 50};
4     for(int i = 0; i < 5; i++) {
5         printf("%d ", arr[i]);
6     }
7     return 0;
8 }
```

Output:

10 20 30 40 50

Example: Taking Input into an Array

```
1 #include <stdio.h>
2 int main() {
3     int arr[5];
4     printf("Enter 5 numbers:\n");
5     for(int i = 0; i < 5; i++) {
6         scanf("%d", &arr[i]);
7     }
8     printf("You entered:\n");
9     for(int i = 0; i < 5; i++) {
10        printf("%d ", arr[i]);
11    }
12    return 0;
13 }
```

Output:

(Depends on input, e.g., You entered: <input values>)

Sum of Elements in an Array

```
1 #include <stdio.h>
2 int main() {
3     int arr[5] = {2, 4, 6, 8, 10};
4     int sum = 0;
5     for(int i = 0; i < 5; i++) {
6         sum += arr[i];
7     }
8     printf("Sum = %d\n", sum);
9     return 0;
10 }
```

Output:

Sum = 30

Finding Maximum and Minimum Element

```
1 int arr[5] = {5, 12, 3, 27, 9};
2 int max = arr[0];
3 int min = arr[0];
4 for(int i = 1; i < 5; i++) {
5     if(arr[i] > max)
6         max = arr[i];
7     if(arr[i] < min)
8         min = arr[i];
9 }
10 printf("Max = %d\n", max);
11 printf("Min = %d\n", min);
```

Output:

Max = 27
Min = 3

Common Mistakes

- ▶ (a) Out-of-Bounds: `arr[5] = 10; // Wrong for size 5`
- ▶ (b) Uninitialized: `int arr[5]; printf("%d", arr[2]); // Garbage`

Visualizing Out-of-Bounds Access

Accessing `arr[5]` on a size 5 array is invalid (highlighted in red).

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| 10 | 20 | 30 | 40 | 50 | |
| [0] | [1] | [2] | [3] | [4] | [5] |

Practice Problems

1

Write a program to read 10 integers and print them in reverse order.

2

Write a program to count the number of even and odd numbers in an array.

3

Write a program to find the average of elements in an array.

4

Write a program to copy one array into another array.

5

Write a program to check if a given number exists in the array (linear search).

6

Write a program to find the second largest number in an array.

Thank You

Keep practicing — happy coding!