Managing Memory with Arrays

Zachary Bennett
SOFTWARE ENGINEER

@z_bennett_   github.com/zbennett10

Array

A container that houses elements of a single type within a contiguous block of memory.

Array Benefits

Arrays are essential for operating on sequential elements of the same type.

Fast Access

Quick read and write operations

Memory Efficient

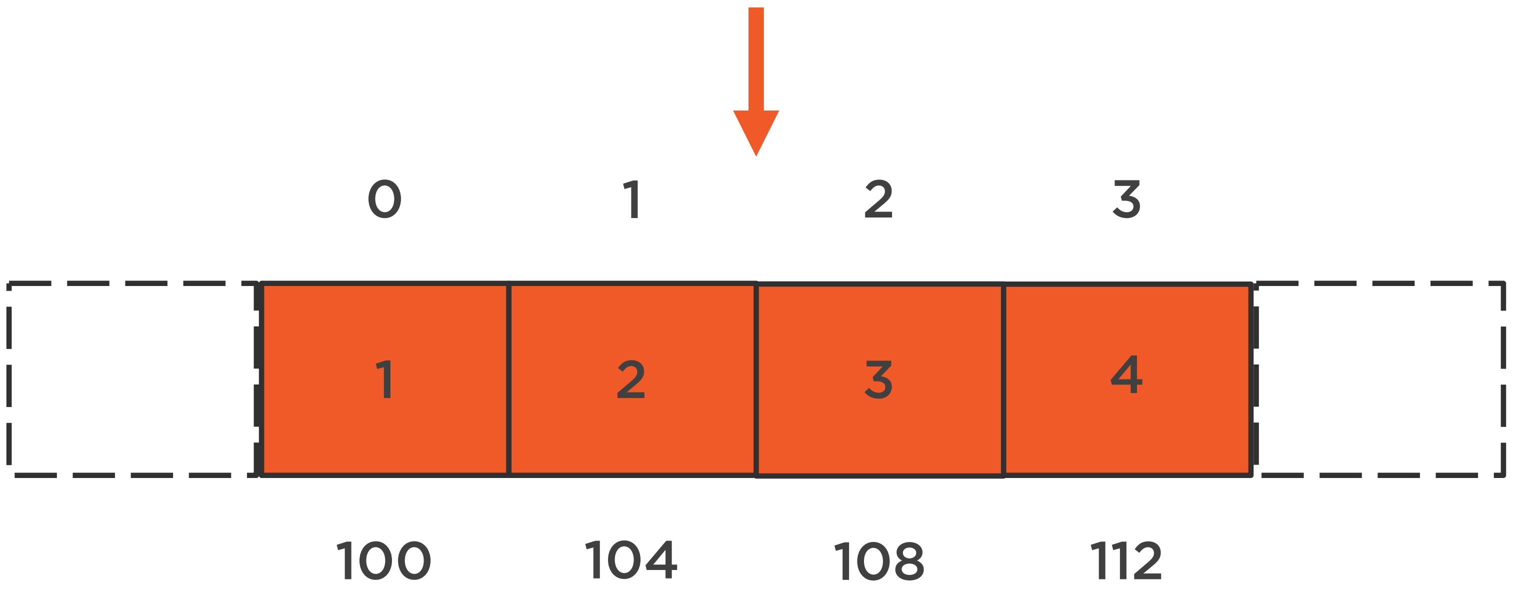
Elements are contiguous within memory

Multidimensional

You can use them to model matrix/record operations

Contiguous Blocks of Memory

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



Arrays vs. Pointers

```
void iterateThroughArray(int *arr, size_t size);

int my_array[] = { 1, 2, 3, 4 };

iterateThroughArray(my_array, 4);
```

Arrays *decay* to pointers

my_array == &my_array[0]

Demo

Declare and initialize one-dimensional arrays

Array notation for reading and writing

Array names decay to pointers

Multidimensional Arrays

Downloads	Age
245	38
78	27
89	26
25	43
10	55

Multidimensional Arrays

```
int arr[][2] = {
    { 245, 38 },
    { 78, 27 },
    { 89, 26 },
    { 25, 43 },
    { 10, 55 }
};
```

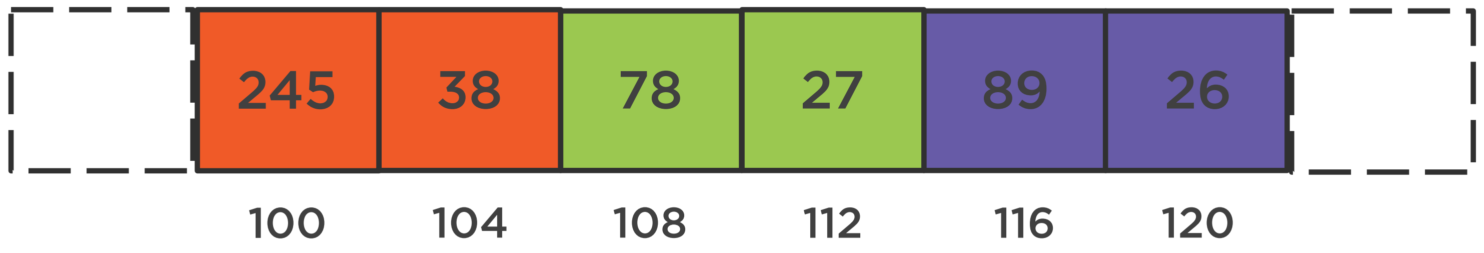
Downloads	Age
245	38
78	27
89	26
25	43
10	55

```
arr[0][0] == 245
arr[0][1] == 38
```

Multidimensional Array Memory Layout

```
int arr[][2] = {
    { 245, 38 },
    { 78, 27 },
    { 89, 26 }
};
```

Row Major Layout



When to Use Multidimensional Arrays

Tabular data

Efficient in-memory storage

Matrix calculations

Demo

Declare and initialize a 2-D array

Iterate over the multi-dimensional array

Read and write values

Dynamically Allocated Arrays

Stored on the heap

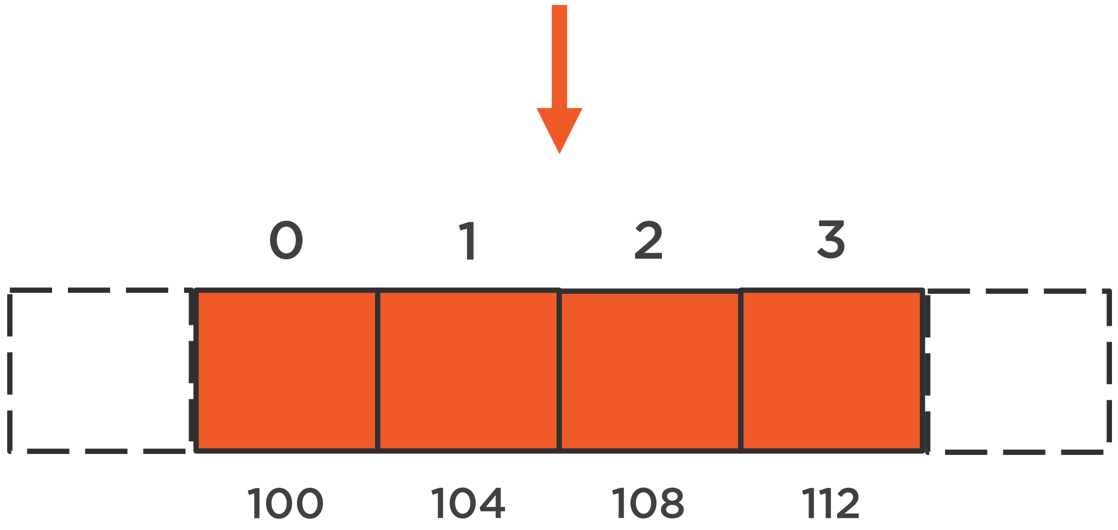
Size is determined at runtime

Resizing is possible

No need to over-allocate memory – useful for large arrays

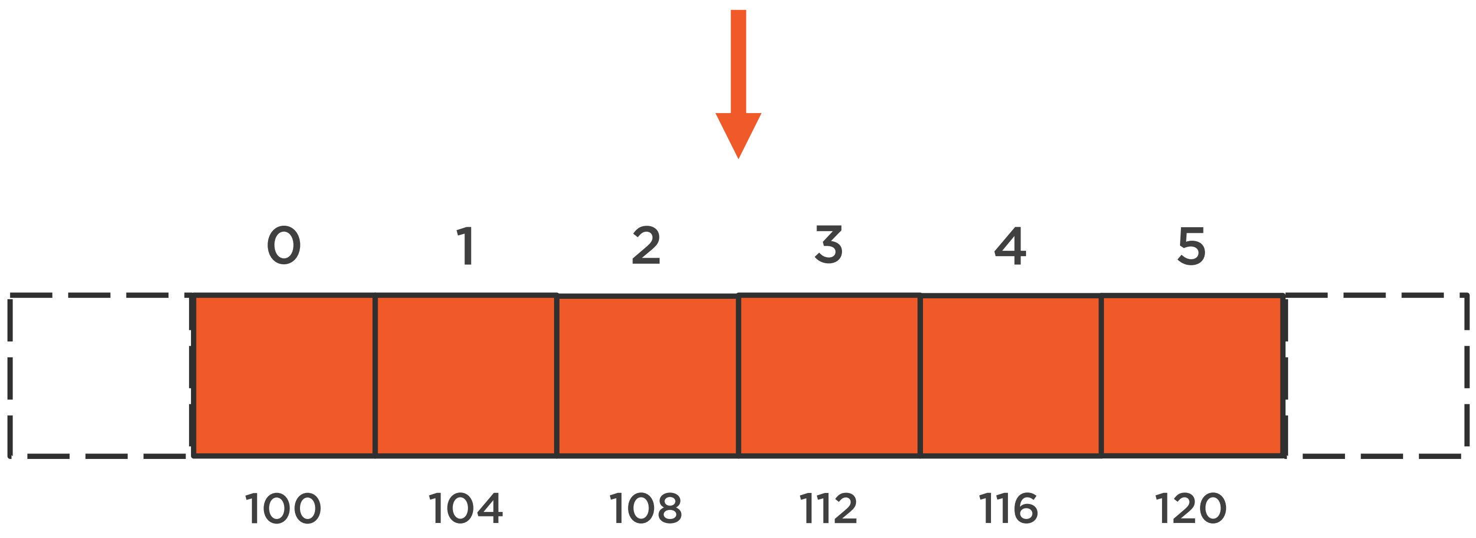
Dynamically Allocated Array Syntax

```
int *arr = (int*)malloc(4 * sizeof(int));
```



Resizing Arrays

```
int *arr = (int*)realloc(arr, 6 * sizeof(int));
```

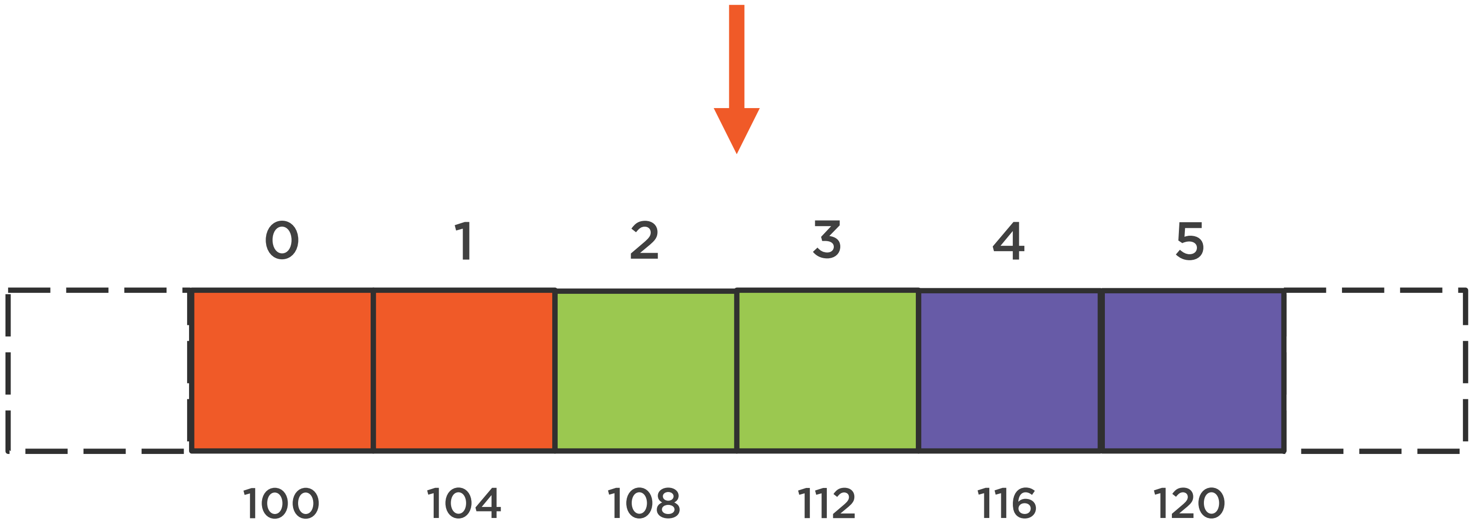


A Quick Note: Variable-length Arrays

```
void create_vla_example(size_t size) {
    double my_array[size];
    ...
}
```

Dynamic Multidimensional Arrays

```
int (*arr)[rows][columns] = malloc( sizeof(int[rows][columns]) );
```



Wired Brain Coffee

Stack Overflow!

Demo

Diagnose problematic variable-length arrays

Replace with dynamically allocated arrays

Provide safe fallbacks

Managing pointers with an array

Overview/ Summary

Sequential elements of a single type:

- Character/Integer/Double/etc.
- Custom types

Indexed

Contiguous in memory

Multidimensional

Statically vs. dynamically allocated