



Data Structures and Algorithms (ES221)

Pointers Arithmetic

Dr. Zubair Ahmad

Pointer and Array

An array name acts like a pointer to its first element

Use pointers to access and manipulate array elements

```
int arr[5] = {10, 20, 30, 40, 50};
```

This means **arr** holds the memory address of **arr[0]**

The array name **arr** is a constant pointer to the first element (**arr[0]**).

Pointer and Array

```
int myarray [20];  
int * mypointer;
```

Which statement is
valid?



`mypointer =myarray;`
`myarray =mypointer;`

Pointer and Array (Example)



Lets code few
examples

Pointer and String

```
char str[] = "World";
```

Automatically includes a null character (`\0`) at the end.

strings can be manipulated using **character arrays** or **pointers to characters**.

The name `str` is a **pointer** to the first character.

```
char *ptr = str; // ptr now points to 'W'
```

```
cout << *(ptr + 1); // Outputs 'o'
```

Pointer and String (example)



Lets code it

Pointer to Pointer (Double Pointer)

A pointer that stores the **address of another pointer**, rather than a direct variable address

In simple terms:

- A **single pointer (*p)** stores the address of a variable.
- A **double pointer (**pp)** stores the address of a single pointer.

```
char a;  
char * b;  
char ** c;  
a = 'z';  
b = &a;  
c = &b;
```

Pointer to Pointer (Double Pointer)

Linked lists, trees,
and other
complex data
structures

Dynamic
memory
allocation in 2D
arrays

Why Use Double Pointers?

Passing a pointer
by reference to
modify it inside a
function

Pointer to Pointer (Double Pointer)

A blue thought bubble with a black outline, containing the text "Lets code it" in white. The bubble has a small tail pointing towards the bottom left.

Lets code it

Pointer to Function

Useful for scenarios like callback functions, function tables, and dynamic function selection

A **pointer to a function** allows us to store the address of a function and invoke it indirectly

Pointer to Function (Examples)

Lets code few examples

Pointer and Structures

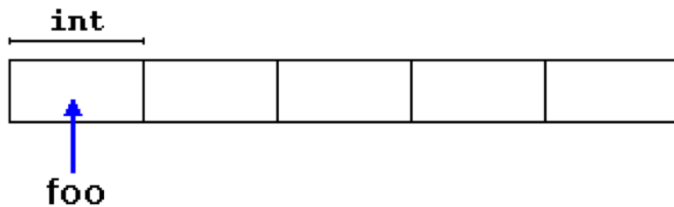
user-defined data
type that groups
multiple variables (of
different types) under
one name

Lets code few examples

Dynamic memory allocation

```
int * foo;  
foo = new int [5];
```

In this case, the system dynamically allocates space for five elements of type `int` and returns a pointer to the first element of the sequence, which is assigned to `foo` (a pointer). Therefore, `foo` now points to a valid block of memory with space for five elements of type `int`.



where the
memory needs of
a program can
only be
determined
during runtime

```
delete pointer;  
delete[] pointer;
```

Dynamic memory allocation



Lets code few examples

Questions?

zahmaad.github.io