

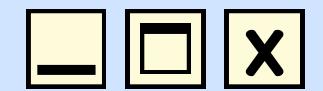
# Embedded Systems

## Task 5

by Mohammed Elahmady



# Task I Objects

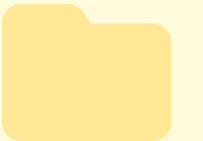


## Task I Objects

> Double Pointer

> Relation between array &  
pointer

> Function Pointer



Double Pointer



Relation between  
array & pointer

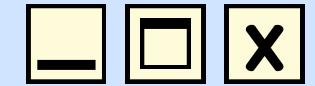


Function Pointer





# Double Pointer



## ◆ 1. Purpose of Double Pointers (`int**`)

A double pointer (pointer to pointer) stores the address of another pointer.

They are mainly used in:

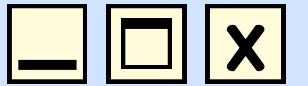
- Dynamic memory allocation (e.g., arrays of pointers)
- Functions that modify pointer arguments
- 2D arrays or lists of strings

```
C test.c > main()
1 void allocateMemory(int** ptr) {
2     *ptr = (int*)malloc(sizeof(int)); // Allocates memory
3     **ptr = 42;
4 }
5
6 int main() {
7     int* p = NULL;
8     allocateMemory(&p); // pass address of pointer
9     printf("%d\n", *p); // Output: 42
10    free(p);
11 }
```





## Relation between array & pointer



### 2. Relation between Pointers, Arrays, and Strings

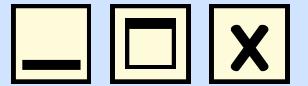
- Arrays and pointers are closely related. The name of an array acts like a pointer to its first element.
- Strings in C are arrays of characters ending with a null character '\0'.
- A char\* can point to a string (character array).

```
C test.c > ...
1  char str[] = "Hello";
2  char* ptr = str;
3
4  printf("%c\n", *(ptr+1));
```





# Function Pointer



## 3. Purpose of Pointer to Function

A function pointer stores the address of a function and is used to:

- Pass functions as arguments (callbacks)
- Create function tables (e.g., in drivers or state machines)
- Allow dynamic selection of functionality

```
C test.c > void greet() {  
1   void greet() {  
2   |   printf("Hello!\n");  
3   }  
4  
5   void runFunction(void (*fptr)()) {  
6   |   fptr(); // Call the function using pointer  
7   }  
8  
9   int main() {  
10  |   runFunction(greet); // Output: Hello!  
11  }
```



# THANK YOU

Head : Tasnem Sabry

Vice : Ahmed Yasser