

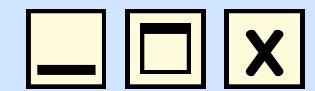
Embedded Systems

Task 6

by Mohammed Elahmady

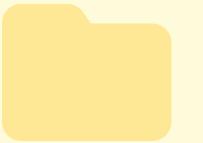


Task 6 Objects

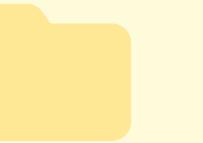


Task 6 Objects

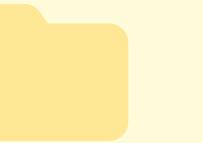
- › Pointers and Arrays in Structures
- › Passing Structures to a Function
- › Size of Structure
- › Memory Padding, Aligned Memory, and Unaligned Memory
- › Difference Between Structures and Objects (Theoretical)



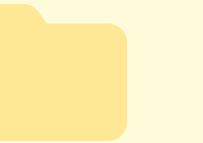
Pointers and Arrays
in Structures



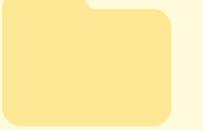
Passing Structures to
a Function



Size of Structure



Memory Padding,
Aligned Memory, and
Unaligned Memory

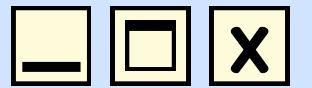


Difference Between
Structures and
Objects (Theoretical)





Pointers and Arrays in Structures



1. Pointers and Arrays in Structures

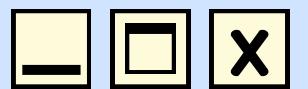
In C, structures can contain pointers and arrays as members.

```
C test.c > ...
1 struct Student {
2     char name[50];      // Array
3     int* marks;         // Pointer
4 }
```





Pointers and Arrays in Structures



2. Passing Structures to a Function

Structures can be passed in three ways:

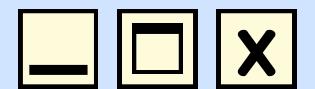
1. By value (copy of the structure)
2. By reference using pointer
3. By reference using const struct* (for read-only)

```
C test.c > ...
1 void printStudent(struct Student s);           // by value
2 void updateMarks(struct Student* s);           // by pointer
```





Pointers and Arrays in Structures



3. Size of Structure

The size of a structure in C is not just the sum of sizes of its members due to memory alignment and padding.

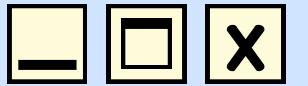
```
C test.c > ...
1 struct Example {
2     char a;      // 1 byte
3     int b;       // 4 bytes
4 };
```

// Size = 8 bytes (not 5) due to padding





Pointers and Arrays in Structures



4. Memory Padding, Aligned Memory, and Unaligned Memory

Concept :

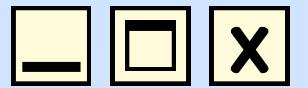
Term	Explanation
Padding	Extra bytes added to align data to word boundaries (e.g., 4 or 8 bytes)
Aligned Memory	Data stored at memory addresses that match CPU alignment rules (fast access)
Unaligned Memory	Data stored without alignment; may cause performance penalties or faults

```
C test.c > ...
1 struct Test {
2     char a; // 1 byte
3     // 3 bytes padding
4     int b; // 4 bytes
5 }; // total = 8 bytes
```





Pointers and Arrays in Structures



5. Difference Between Structures and Objects (Theoretical)

Concept :

Concept	Structures (C)	Objects (OOP, C++/Java)
Encapsulation	Grouping of data only	Encapsulates both data and functions
Access Control	No built-in access specifiers (public only)	Has public, private, protected
Inheritance	Not supported	Supported
Polymorphism	Not supported	Supported
Memory Model	Simple static/dynamic	Complex with constructors, destructors, vtables, etc.



THANK YOU

Head : Tasnem Sabry

Vice : Ahmed Yasser