C Programming Cheat Sheet (Structures & Pointers) – Theory Only

---

Structures in C

Definition & Declaration of Structures

- A **structure** is a user-defined data type that groups related variables of different data types.
- **Structure Definition**:
- `struct Student {`
- `    char name[50];`
- `    int roll_no;`
- `    float marks;`
- `};`
-
- **Structure Declaration & Initialization**:
- `struct Student s1 = {"Alice", 101, 89.5};`
-

Structure within a Structure (Nested Structure)

- A structure can contain another structure as a member.
- `struct Address {`
- `    char city[20];`
- `    int pin;`
- `};`
-
- `struct Student {`
- `    char name[50];`
- `    struct Address addr;`
- `};`
-

Self-Referential Structure

- A structure that contains a pointer to itself.
- Used in **linked lists, trees, and graphs**.
- `struct Node {`
- `    int data;`
- `    struct Node *next;`
- `};`
-

Pointers in C

Definition & Initialization

- A **pointer** is a variable that stores the **memory address** of another variable.
- **Declaration**: int *ptr; (Pointer to an integer)
- **Initialization**:
- `int a = 10;`
- `int *ptr = &a; // Stores address of 'a'`

-

Pointer Arithmetic

- **Increment (ptr++)** → Moves to next memory location.
- **Decrement (ptr--)** → Moves to previous location.
- **Addition (ptr + n)** → Moves n locations forward.
- **Subtraction (ptr - n)** → Moves n locations backward.

Pointers & Arrays

- A pointer can be used to access array elements.
- `int arr[5] = {1, 2, 3, 4, 5};`
- `int *ptr = arr; // Points to first element`
- `printf("%d", *(ptr + 2)); // Prints 3`
-

Pointer to a Function

- **Used to call a function dynamically.**
- `void greet() {`
- `    printf("Hello, World!");`
- `}`
-
- `void (*func_ptr)(); // Function pointer declaration`
- `func_ptr = greet; // Assign function to pointer`
- `func_ptr(); // Calls greet()`
-

Pointer & Structure

- **Accessing structure members using pointers**.
- `struct Student {`
- `    char name[50];`
- `    int roll_no;`
- `};`
-
- `struct Student s1 = {"Bob", 102};`
- `struct Student *ptr = &s1;`
-
- `printf("%s", ptr->name); // Access using '->' operator`
-

This **C Programming Cheat Sheet** covers **structures, pointers, and their applications**. Let me know if you need more details!