C Programming Cheat Sheet (Unions, File Handling, Storage Classes, Pre-Processor, Dynamic Memory) – Theory Only

---

Unions in C

Introduction to Unions

- A **union** is a special data type that allows storing different data types in the same memory location.
- Unlike structures, **a union shares the same memory for all members**, meaning only one value can be stored at a time.

Union Declaration & Usage

```
union Data {
    int i;
    float f;
    char str[20];
};

union Data d1; // Declaration
```

Difference Between Structure & Union

Feature	Structure	Union
Memory	Allocates separate memory for each member	All members share the same memory
Access	All members can be accessed at once	Only one member can store a value at a time
Use Case	Storing multiple related values	Saving memory when storing different values at different times

---

File Handling in C

Introduction to Files

- A **file** is used to store data permanently.
- **Types of Files**:
- **Text Files** → Store data in human-readable format.
- **Binary Files** → Store data in machine-readable format.

File Operations

1st **Opening a File** → fopen("filename", "mode")
2nd**Reading from a File** → fscanf(), fgets(), fread()
3rd**Writing to a File** → fprintf(), fputs(), fwrite()
4th**Closing a File** → fclose()

File Modes

Mode	Description
"r"	Read mode (file must exist)
"w"	Write mode (creates a new file or overwrites existing)
"a"	Append mode (adds data to the end of the file)
"r+"	Read and write
"w+"	Write and read (overwrites file)
"a+"	Append and read

Random Access to Files

- Used to **read/write specific positions** in a file.
- **Functions**:
- fseek(fp, offset, position) → Moves the file pointer.
- ftell(fp) → Returns the current position.
- rewind(fp) → Moves the pointer to the start of the file.

File System Functions

- remove("filename") → Deletes a file.
- rename("oldname", "newname") → Renames a file.

Command Line Arguments

- Used to pass inputs while executing a program.
- **Syntax in main()**:

```
int main(int argc, char *argv[]) {
    printf("Argument Count: %d", argc);
    printf("First Argument: %s", argv[0]); // Program name
    return 0;
}
```

Storage Classes in C

Storage Class	Scope	Lifetime	Default Value
auto	Local	Function execution	Garbage
register	Local (Stored in CPU register)	Function execution	Garbage
static	Local/Global	Entire program	0 (Numeric), NULL (Pointer)
extern	Global (Declared in another file)	Entire program	0 (Numeric)

Preprocessor Directives

- Preprocessor directives are **commands that run before compilation**.

- **Common Preprocessor Directives**:
- #include → Includes a header file.
- #define → Defines a macro.
- #ifdef / #ifndef → Conditional compilation.

Example:
```
#define PI 3.14159
```

Dynamic Memory Allocation

- **Used to allocate memory at runtime.**
- **Functions in <stdlib.h>**:
- malloc(size) → Allocates memory but does not initialize.
- calloc(n, size) → Allocates and initializes memory with zeros.
- realloc(ptr, new_size) → Resizes allocated memory.
- free(ptr) → Deallocates memory.

Example:
```
int *ptr = (int*)malloc(10 * sizeof(int)); // Allocates memory for 10 integers
free(ptr); // Frees memory
```

This **C Programming Cheat Sheet** covers **unions, file handling, command-line arguments, storage classes, preprocessor directives, and dynamic memory allocation**. Let me know if you need more details!