

File Handling in C

1. File Pointer

- In C, files are handled using a **file pointer** of type `FILE`.
- Declaration:
- `FILE *fp;`
- The file pointer keeps track of the file being used.

2. Opening a File: `fopen()`

- Used to open a file and associate it with a file pointer.
- Syntax:
- `FILE *fopen(const char *filename, const char *mode);`
- Returns:
 - **File pointer** if successful
 - **NULL** if file cannot be opened

Example:

```
FILE *fp;  
fp = fopen("data.txt", "r");
```

3. Closing a File: `fclose()`

- Used to close an opened file.
- Syntax:
- `int fclose(FILE *fp);`
- Always close files after use to avoid memory/data loss.

Example:

```
fclose(fp);
```

4. Reading & Writing with `fprintf` and `fscanf`

(a) `fprintf()` → Write to File

- Similar to `printf` but writes into a file.
- Syntax:
- `int fprintf(FILE *fp, const char *format, ...);`

Example:

```
fprintf(fp, "Name: %s Age: %d\n", name, age);
```

(b) `fscanf()` → Read from File

- Similar to `scanf` but reads from a file.
- Syntax:

- `int fscanf(FILE *fp, const char *format, ...);`

Example:

```
fscanf(fp, "%s %d", name, &age);
```

5. File Modes in C

Mode	Meaning	Behavior
"r"	Read	Opens existing file for reading. File must exist.
"w"	Write	Creates a new file for writing. If file exists, contents are erased.
"a"	Append	Opens file for writing. Data is added at end. Creates file if it doesn't exist.
"r+"	Read + Write	Opens file for both reading and writing. File must exist.
"w+"	Write + Read	Creates new file for reading & writing. If file exists, contents are erased.
"a+"	Append + Read	Opens file for reading & appending. Creates file if it doesn't exist.

Quick Example (Write & Read):

```
#include <stdio.h>

int main() {
    FILE *fp;
    char name[20];
    int age;

    // Writing to file
    fp = fopen("data.txt", "w");
    fprintf(fp, "Neeraj 23\n");
    fclose(fp);

    // Reading from file
    fp = fopen("data.txt", "r");
    fscanf(fp, "%s %d", name, &age);
    printf("Name: %s, Age: %d\n", name, age);
    fclose(fp);

    return 0;
}
```