13. File Handling in C

THEORY EXERCISE:

- Explain the importance of file handling in C. Discuss how to perform file operations like opening, closing, reading, and writing files.

Why File Handling is Important:

- **Data storage**: Keeps data even after the program ends.
- **Large data management**: Handles large amounts of input/output easily.
- **Data sharing**: Allows reading/writing data from/to external files.
- **Automation**: Useful in reading from or writing to logs, configuration files, etc.
  - File handling in C is crucial because it enables programs to interact with the file system, allowing them to store data persistently, read from and write to files, and perform other file-related operations. This is essential for various applications, such as storing user data, saving program logs, or managing configuration files.

Modes:

Mode	Meaning
"r"	Read (file must exist)
"w"	Write (create new or overwrite)
"a"	Append (write at end)
"r+"	Read + Write
"w+"	Read + Write (overwrite file)
"a+"	Read + Append

file operations in C:

1. Opening a File:

- The fopen() function is used to open a file.

- It takes two arguments: the file path and the mode (e.g., "r" for read, "w" for write, "a" for append).
- For example, FILE *fp = fopen("my_file.txt", "w"); opens a file named "my_file.txt" for writing.

2. Reading from a File:

- Several functions are available for reading data from a file, including fscanf(), fgetc(), fgets(), and fread().
- fscanf() reads formatted data from a file, similar to scanf() but for files.
- fgetc() reads a single character from the file, and fgets() reads a line of text.
- fread() is used for binary file reading.

3. Writing to a File:

- Functions like fprintf(), fputc(), fputs(), and fwrite() are used for writing data to a file.
- fprintf() writes formatted data to a file, similar to printf().
- fputc() writes a single character to the file, and fputs() writes a string.
- fwrite() is used for binary file writing.

4. Closing a File:

- The fclose() function is used to close a file after it's been opened and used.
- It takes the file pointer as an argument.
- For example, fclose(fp); closes the file associated with the file pointer fp.