UNIT 11
DATA FILE HANDLING
LH - 4HRS

PRESENTED BY: **ER. SHARAT MAHARJAN**

C PROGRAMMING

PRIME COLLEGE, NAYABAZAAR

CONTENTS (LH - 4HRS)

11.1 Introduction

11.2 Types of File

11.3 Opening and Closing Data File

11.4 Read and Write Function

11.5 Writing and Reading Data to and From Data File

11.6 Updating Data File

11.7 Random Accessing Files

11.8 Printing and File

11.1 Introduction

- Programs without data files accept input data from the keyboard at the time of execution and write to the monitor.
- This type of I/O is called console I/O.

Problems with console I/O:

- Entire data is lost when either the program is terminated or the computer is turned off.
- When the volume of data to be entered is large, it takes a lot of time to enter the data.
- If user makes a mistake while entering data, whole data has to be re-entered.

Solution: File

File handling in C enables us to **create, update, read, and delete the files** stored on the local file system through our C program. The following operations can be performed on a file.

- Creation of the new file
- Opening an existing file
- Reading from the file
- Writing to the file
- Closing the file

11.2 Types of File

- Generally, a file is used to store user data in a computer.
- File is a **collection of data** stored on secondary memory like hard disk of a computer.
- C programming language supports **two types of files** and they are as follows:

✓**Text File (or) ASCII File** - The text file contains textual information in the form of alphabets, digits and special characters or symbols.

✓**Binary File** - The file that contains data in the form of bytes (0's and 1's) is called as binary file. Generally, the binary files are compiled version of text files.

11.3 Opening and Closing Data File

- To create a new file or open an existing file, we need to create a file pointer of FILE type.
- Following is the sample code for creating **file pointer**.

FILE *fp ;

- We use the pre-defined method **fopen()** to **create a new file or to open an existing file**.

File *fp ;

fp = fopen(“C:\\abc.txt", "w") ;

- The **fclose()** function is used to close a file.
- The syntax of fclose() function is given below:

fclose(fp);

LAB 1: WAP to create a file named test.txt and write some text “I study CSIT”.

```
#include<stdio.h>
#include<stdlib.h>
int main(){
        FILE *fp;
        fp=fopen("D:\\test.txt","w");        //write-mode
        if(fp==NULL){
                printf("\n File can't be created.");
                exit(0);
        }
        fputs("I study CSIT.", fp);
        fclose(fp);
        return 0;
}
```

LAB 2: WAP to open the file named test.txt, read its content and display it to screen.

```
#include<stdio.h>
#include<stdlib.h>
int main(){
        FILE *fptr;
        char message[100];
        fptr = fopen("D:\\test.txt","r");          //read-mode
        if(fptr==NULL){
                printf("No file found.");
                exit(0);
        }
        fgets(message,20,fptr);
        printf("%s",message);
        fclose(fptr);
        return 0;
}
```

LAB 3: WAP to open the file named test.txt and add to it the text "At Prime College".

```
#include<stdio.h>
int main(){
        FILE *fptr;
        fptr = fopen("D:\\test.txt",“a");  //append-mode
        if(fptr==NULL){
                printf("File can't be opened.");
        }
        fputs("At Prime College.",fptr);
        fclose(fptr);
        return 0;
}
```

LAB 4: WAP to open a file and copy all its content to another file.

```
#include<stdio.h>
#include<stdlib.h>
int main(){
        FILE *sfptr,*dfptr;
        char c;
        sfptr=fopen("D:\\test.txt","r");
        if(sfptr==NULL){
                printf("File can't be opened.");
                exit(0);
        }
        dfptr=fopen("D:\\dest.txt","w");
        if(dfptr==NULL){
                printf("File can't be created or opened.");
                exit(0);
        }
        while((c=fgetc(sfptr))!=EOF){
                fputc(c,dfptr);
        }
        fclose(sfptr);
        fclose(dfptr);
        return 0;
}
```

THANK YOU FOR YOUR ATTENTION