Programme : Bachelor of Science (First Year)

Subject : Computer Science

Semester : I

Course Code : CSC101

Course Title : Programming Fundamentals Using C

Title of the Unit : File Handling

Module Name : Text and Data file create, open

What is a FILE?

•A file is a stream of bytes stored on some secondary storage devices.

Typical operations on files:

- Open
- Read
- Write
- Close

Why files are needed?

- When a program is terminated, the entire data is lost. Storing in a file will preserve data even if the program terminates.
- It is time consuming to enter a large number of data,However, using a file containing all the data, makes it easier to access the contents of the file using a few commands in C.
- Easier to move data from one computer to another without any changes.

Types of Files

1. Text files

- Text files are the normal **.txt** files.
- Can easily create text files using any simple text editors such as Notepad.
- All the contents within the file is stored as plain text.
- Requires minimum effort to maintain, are easily readable, and provide the least security and takes bigger storage space.

Binary files

- Binary files are mostly the **.bin** files.
- Instead of storing data in plain text, it is stored in binary form (0's and 1's).
- Binary files can hold a higher amount of data, are not readable easily, and provides better security than text files.

How is a file stored?

- Stored as sequence of bytes, logically contiguous (may not be physically contiguous on disk).
- The last byte of a file contains the end-of-file character (EOF).
- While reading a text file, the EOF character can be checked to know the end.

File Opening Modes

Mode	Meaning of Mode	During Inexistence of file
r	Open for reading	If the file does not exist, fopen() returns NULL.
rb	Open for reading in binary mode.	If the file does not exist, fopen() returns NULL.
w	Open for writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb	Open for writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a	Open for append. Data is added to the end of the file.	If the file does not exist, it will be created.
ab	Open for append in binary mode. Data is added to the end of the file.	If the file does not exist, it will be created.
r+	Open for both reading and writing.	If the file does not exist, fopen() returns NULL.
rb+	Open for both reading and writing in binary mode.	If the file does not exist, fopen() returns NULL.
w+	Open for both reading and writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.

wb+	Open for both reading and writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a+	Open for both reading and appending.	If the file does not exist, it will be created.
ab+	Open for both reading and appending in binary mode.	If the file does not exist, it will be created.

File Management Functions

fopen()	opens new or existing file
fprintf()	write data into the file
fscanf()	reads data from the file
fputc()	writes a character into the file
fgetc()	reads a character from file
fclose()	closes the file
fseek()	sets the file pointer to given position
fputw()	writes an integer to file
fgetw()	reads an integer from file
ftell()	returns current position
rewind()	sets the file pointer to the beginning of the file

FILE OPERATIONS

1) File Pointer

In C we use FILE * to represent a pointer to a file.

- fopen() is used to open a file.
- It returns the special value NULL to indicate that it is unable to open the file.

SYNTAX:

FILE * <identity>;

Eg: FILE * fp;

Opening a file- for creation and edit

SYNTAX:

FILE * fopen(“filename”,”mode”);

- fopen() function, is defined in <stdio.h> header file
- The name of the file to be opened is pointed to by *filename*

- The string given as the second parameter - for mode, determines how the file should be accessed (r-read, w-write a-append).

Example:

- fopen("C:/Users/Lenovo/Documents/file1.txt","w");
- fopen("C:/Users/Lenovo/Documents/file2.bin",“rb");

Closing a file

SYNTAX:

fclose(FILE * fp);

- The file (both text and binary) should be closed after reading/writing.
- fclose() returns 0 if the file is closed successfully

Example:

fclose(fp);

Example :

Program to create and open a text file.

```
#include<stdio.h>
main()
{
FILE *fp;
fp=fopen("file.txt","r");
if(fp ==NULL)
printf("File not present \n");
else
  printf("File opened in read mode \n");
}
```