File Management in C

The console oriented I/O operations pose two major problems.

- ☆ It becomes time consuming to handle large volumes of data through terminals.
- ☆ The entire data is lost when either the Program is terminated or the computer is turned off.

FILE :- A file is a place on the disk where a group of related data is stored. C supports a number of functions that have the ability to perform basic file operations, which include:

naming a file,
opening a file,
reading data from a file,
writing data to a file and
closing a file.

Function Name	operation
fopen()	creates a new file for use Opens an existing file for use.
fclose()	closes a file which has been opened for use.
getc()	Reads a character from a file
putc()	writes a character to a file
fprintf()	writes a set of data values to a file.
fscanf()	Reads a set of data values from a file
getw()	Reads an integer from a file
putw()	writes an integer to a file
fseek()	Sets the position to a desired point in the file.
ftell()	Gives the current position in the file (in terms of bytes from the start)
rewind()	Sets the position to the begining of the file

Defining And opening a File :- Following is the general format for declaring and opening a file:

```
FILE *fp;
fp = fopen("filename", "mode");
```

The first statement declares the variable fp as a "pointer to the data type File". File is a structure that is defined in the I/O library. The second statement opens the file named filename and assigns an identifier to the FILE type pointer fp.

MODE :-

r open the file for reading only.

w open the file for writing only

a open the file for appending (or adding) data to it.

r+ The existing file is opened to the beginning for reading and writing.

w+ Same as w except both for reading and writing.

a+ Same as a except both for reading

CLOSING A FILE :-

```
fclose(file-pointer);
```

This would close the file associated with the file pointer

INPUT / OUTPUT OPERATIONS ON FILES :-

```
putc(character_variable name, filepointer);
```

writes the character contained in the character variable to the file associated with filepointer.

```
character variable = getc(file pointer);
```

would read a character from the file & store it into the character variable.

NOTE :- The file pointer moves by one character position for every operation of getc or putc.

The getc will return an end of file marker EOF, when end of the file has been reached. Therefore, the reading should be terminated when EOF is encountered.

ex:-
```
while ((c=getc(fp)) !=EOF)
```

fprintf and fscanf functions :-

```
fprintf ( file pointer, "control string", list);

fscanf ( file pointer, "control string, list);

while ( ! feof(fp))
     {

     }
```