CHAPTER NO: 8	FILE HANDLING
1	Explain fopen() with all its modes.
2	Explain fclose() and use of that function.
3	Explain fgetc() and fputc() with example.
4	Explain fprintf() and fscanf() with example .
5	Explain ftell(), fseek() and rewind() with example.
6	Explain fgets() and fputs() with example.
7	Explain command line arguments with example.

Q-1 EXPLAIN FOPEN() WITH ALL ITS MODES.

ANS:

- fopen() is used for opening the file.

  Syntax:

  fopen("filename",filemode")

- The first argument (filename) contains the name of the file and the extension of file.
  e.g. 1.txt

- The second argument (filemode) are of 6 types:

  1) **"r" (read mode):-**
  This mode searches the file. If the file exists then it loads in to memory and allows you to read the data of existing file. If the file doesn't exist then it will return NULL value that is it will not create new file.

  2) **"w" (write mode):-**
  This mode searches the file. If the file exists then it loads into memory and allows you to write new data. The data of existing file is erased and if the file doesn't exist then it will create new file.

  3) **"a" (append mode):-**
  This mode searches the file. If the file exits then it loads into memory and allows you to append. This mode will allow you to read the existing data and then append new data. If the file doesn't exist then it will create new file.

4) **“r+”(read, write mode) :-**

This mode searches the file. If the file exists then it loads into memory and allows you to read the existing data and write the new data in file. If the file doesn’t exist then it will not create new file.

5) **“w+” (write, read mode):-**

This mode searches the file. If the file exists then it loads into memory and allows you to write new data and read that data. The data of existing file are erased. If the file doesn’t exist then it will create new file.

6) **“a+”(read, write, append, modify, existing data):-**

This mode searches the file. If the file exists then it loads into memory and allows you to read, write, append. If the file doesn’t exist then it will create new file.

Note:

The difference between r+ and w+ is that:-

1) In r+ mode, if the file does not exist then it will not create new file while in w+ mode, if the file does not exist then it will create new file.
2) In r+ mode, data of existing file are not erased while in w+ mode, data of existing file are erased.

Q-2 EXPLAIN FCLOSE() AND USE OF THAT FUNCTION.

ANS:

- This function is used to close the file.

Syntax:

```
fclose(FILE  *f1);
```

Example :

```
#include<stdio.h>
#include<conio.h>
```

```
void main()
{
        FILE *f1;
        f1=fopen("c:\\1.txt","r");
        if(f1==NULL)
        {
                Printf("\n FILE NOT FOUND");
        }
        fclose(f1);
        getch();
}
```

Q-3 EXPLAIN FGETC() AND FPUTC() WITH EXAMPLE.

ANS:

fgetc():

- fgetc() is used to get or read the character from file.

  Syntax:

  fgetc(ch,f1);

- **In the above syntax, ch is any character.**

  Example :

```
#include<stdio.h>
#include<conio.h>

void main()
{
        FILE *f1,*f2;
        char ch;
        clrscr();
        f1=fopen("c:\\1.txt","r");
        f2=fopen("c:\\2.txt","w");
        if(f1==NULL)
        printf("\n FILE NOT FOUND");
        while(!feof(f1))
        {
                fgetc(ch,f1);
                fputc(ch,f2);
```

```
		}
		fclose(f1);
		fclose(f2);
	getch();
	}
```

fputc():

- This function is used to put the character to the file.
- This function has 2 arguments.

Syntax:

fputc(ch,f1);

- **In the above syntax, ch is any character.**

Example :

```
	#include<stdio.h>
	#include<conio.h>

	void main()
	{
		FILE *f1;
		char ch='A';
		f1=fopen("c:\\1.txt","w");
		fputc(ch,f1);
		fclose(f1);
		getch();
	}
```

Q-4 EXPLAIN FPRITNF() AND FSCANF() WITH EXAMPLE.

ANS:

fprintf():

- This function is used to print the values to the file.
- This function has 3 arguments:-

First argument	:-	file pointer
Second argument t	:-	control string
Third argument	:-	variable name

Syntax:

fprintf(f1, “control string”, variable name);

Example :

```
#include<stdio.h>
#include<conio.h>

void main()
{
        FILE *f1;
        int a=10;
        float n=10.5;
        char ch=’A’;
        clrscr();
        f1=fopen(“c:\\1.txt”,”w”);
        fprintf(f1,”%d%f%c”,a,n,ch);
        fclose(f1);
        getch();
}
```

fscanf():

- This function is used to read the values from stdin (Standard input stream).
- This function has 3 arguments:-

First argument	:-	file pointer
Second argument t	:-	control string
Third argument	:-	variable name

Syntax:

fscanf(f1, “control string”, variable name);

Example :

```
#include<stdio.h>
#include<conio.h>

void main()
{
        FILE *f1;
        int a;
        clrscr();
        f1=fopen(“c:\\1.txt”,”w”);
        printf(“\n enter the value of a”);
        fscanf(“%d”,&a);
        printf(“\n value of a=%d”,a);
        fclose(f1);
        getch();
    }
```

Q-5 EXPLAIN FTELL(), FSEEK() AND REWIND() WITH EXAMPLE.

ANS:

ftell():

- ftell() returns relative offset (in bytes) of current position of file. Offset always starts from 0.

Syntax:

long ftell(f1);

Example :

```
#include<stdio.h>
#include<conio.h>
void main()
{
        FILE *f1;
        clrscr();
        f1=fopen("c:\\rt.txt","w");
        printf(“\n starting position=%d”,ftell(f1));
        fprintf(f1,”hello”);
        printf(“\n Current file position=%d”,ftell(f1));
        fclose(f1);
        getch();
}
```

fseek():

- fseek() is used to set the file position means that it is used to move the file position to a choosen location within the file.

Syntax:

fseek(f1,offset,position);

- In the above syntax, first argument is file pointer.
- Second argument is offset which is a number or variable which specifies number of bytes to be moved from location specified by location.
- Third argument is position which is an integer number. It has following 3 values:-

0 OR SEEK_SET->Beginning of file
1 OR SEEK_CUR-> Current Position
2 2 OR SEEK_END-> End of File.

Example :

```
#include<stdio.h>
#include<conio.h>
void main()
{
        FILE *f1;
        clrscr();
        f1=fopen("c:\\2.txt","w");
        fputs("hello hw r u",f1);
        fseek(f1,5,SEEK_SET);
        fputs("hi",f1);
        fclose(f1);
        getch();
}
```

rewind():

- **This function is used to set the file position to the beginning of file or to the start of file.**

Syntax:

rewind (f1);

Example :

```
#include <stdio.h>
#include <conio.h>

void main()
{
        FILE *f1;
        clrscr();
        f1=fopen("c:\\1.txt","w");
        fprintf(f1,"Hi Hw R U");
        printf("\n Current file position=%d",ftell(f1));
        rewind(f1);
        printf("\n after rewind file position=%d",ftell(f1)));
        fclose(f1);
        getch();
}
```

Q-6 EXPLAIN FPUTS() AND FGETS() WITH EXAMPLE.

ANS:

fputs():

- This function is used to put or print the string to the file.
- This function has 2 arguments.

Syntax:

fputs(char *s,f1);

Example :

```
#include<stdio.h>
#include<conio.h>

void main()
{
        FILE *f1;
        char s[5]="hello";
        f1=fopen("c:\\1.txt","w");
        fputs(s,f1);
        fclose(f1);
        getch();
}
```

fgets():

- This function is used to get the string to the file.
- This function has 3 arguments.

First argument	:-	address where the string will be stored
Second argument	:-	maximum length of string
Third argument	:-	file pointer

Syntax:

```
fgets(char *s, int n, FILE *f1);
```

Example :

```
#include<stdio.h>
#include<conio.h>

void main()
{
        FILE *f1;
        int count=0;
        char s[50];
        if(f1=fopen("1.txt","r")==NULL)
        {
                printf("\n Error  opening file");
                exit(0);
        }
        while(!feof(f1))
        {
                fgets(s,80,f1);
                count++;
        }
        printf("\n number of lines in file is %d",count);
        getch();
}
```

Q-7 EXPLAIN COMMAND LINE ARGUMENTS WITH EXAMPLE.

ANS:

- Command line arguments are the arguments that are passed when the program is executed.
- In command line arguments, main() also takes the arguments as their values.

- main() takes two arguments:
  - argc $\rightarrow$ Number of arguments passed
  - argv[] $\rightarrow$ pointer array which points to each argument which is passed to main().
  -

Note:- argv[0] $\rightarrow$ name of the program that is exevuted.
argv[1] $\rightarrow$ First argument
argv[2] $\rightarrow$ Second argument
.
.

- When we want to execute command line program, at that time we have to follow below steps:

  Step 1: Compile and run program that is you have to create exe of program.
  Step 2: Then go to file menu and select DOS Shell
  Step 3: After that we have to pass first argument as our program name and then after that no. of arguments input as per user requirements.

Example:

```
#include<stdio.h>
#include<conio.h>

void main(int argc, char *argv[])
{
        FILE *f1,*f2;
        char ch;
        if(argc<3)
        printf("\n IMPROPER ARGUMENTS");
        f1=fopen(argv[1],"r");
        f2=fopen(argv[2],"w");
        if(f1==NULL)
        printf("\n FILE NOT FOUND");
        while(1)//always ture
        {
                ch=fgetc(f1);
                if(ch==EOF)
                break;
                else
                fputc(ch,f2);
        }
        fclose(f1);
        fclose(f2);
        getch();
}
```

C Language