56. File Management In C Programming:

Introduction:

File defines a computer resource for recording data in a computer storage device, primarily identified but its file name. just as words can be written to paper. So can data be written to a computer file. files can be shared with and transferred between computers and mobile devices via removable media, networks or the internet.

A file container in a computer system that stores data, information, settings or commands which are used with a computer program. In graphical user interface such as Microsoft operating systems, represent the files as icons, which associate to the program that opens the file. For instance the picture is shown as an icon, it is related to Microsoft word. If your computer contains the file and you double-click on the icon. It will open in Microsoft word installed on the computer. A file can be defined as a collection of characters stored on a permanent storage device. c programming language treats all the devices as files devices as standard output, printer etc. C programming language provides a rich set of built in function to read given input and feed it to the program as per requirement as well as for output.

In programming we may require some specific input data to be generated several numbers of times. Sometimes it is not enough to only display the data on the console. The data to be displayed may be very large, and only a limited amount of data can be displayed on the console, and since the memory is volatile, it is impossible to recover the programmatically generated data again and again.

However, if we need to do so, we may store it onto the local file system which is volatile and can be accessed every time. Here, comes the need of file handling in C. 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.

1. Creation Of The New File 2. Opening An Existing File

3. Reading From The File 4. Writing To The File 5. Deleting The File

The Following files are automatically opened when a program executes to provide access to the keyboard (console input device) and screen (console output device)

Standard input – stdin – keyboard

Standard output – stdout – screen

Standard error – stderr – your screen

Functions For File Handling In C Programming:

These are the following functions that are used in C library to open, read, write, search and close file.

fopen() – open 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

fputs() – writes an String to file

fgets() – reads an String from file

fseek() – sets the file pointer to given position

ftell() – returns current position

rewind() – sets the file pointer to the beginning of the file

fclose() – close the file

Opening File: fopen()

fopen() function is used for open new or existing file. You can use the fopen() function to create a new file or to open an existing file, this call will initialize an object of the type file, which contains all the information necessary to control the stream. “FILE *fopen(const char * filename, const char * mode);” The file name (string). If the file is stored at some specific location, then we must mention the path at which the is stored. The mode in which the files is to be opened. It is a string. The following modes are used in fopen() function.

r = opens a text file in read mode

r+ = open a text file in read and write mode

w = opens a text file in write mode

w+ = open a text file in read and write mode

a = opens a text file in append mode

a+ = open a text file in read and write mode rb = opens a binary file in read mode

rb+ = opens a binary file in read and write mode

wb = opens a binary file in write mode

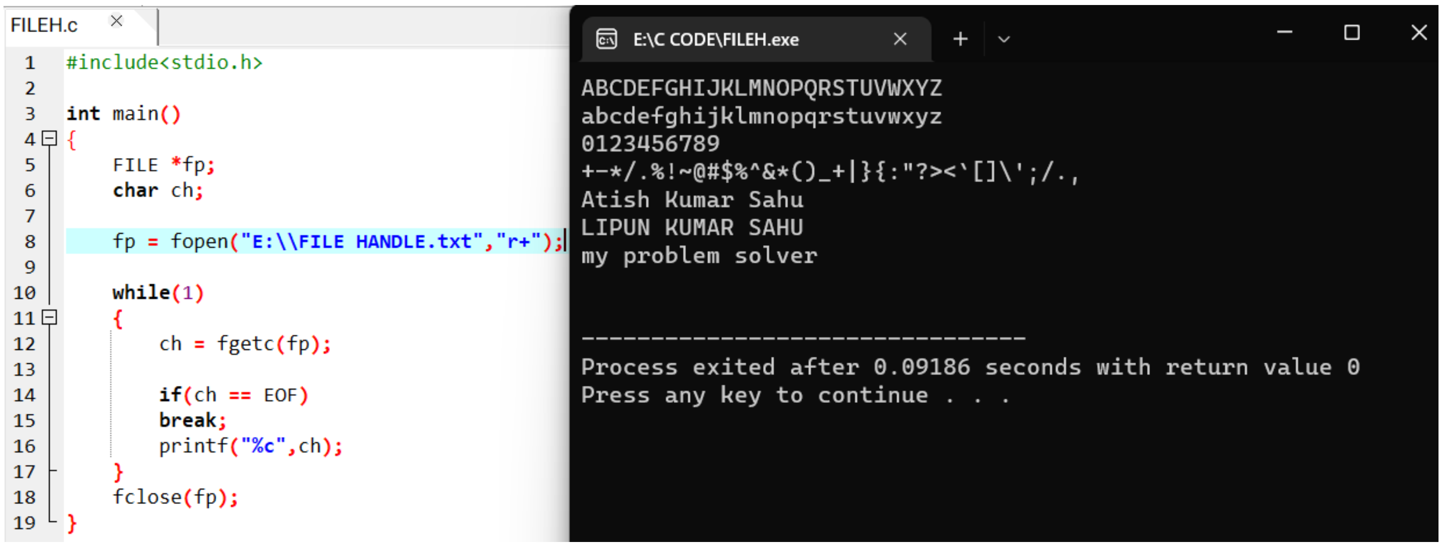
wb+ = opens a binary file in read and write mode

ab = open a binary file in append mode

ab+ = opens a binary file in read and write mode

fopen function works in the following way:

firstly it searches the file to be opened. Then it loads the file from the disk and place it into the buffer. The buffer is used to provide efficiency for the read operations. It sets up a character pointer which points to the first character of the file.

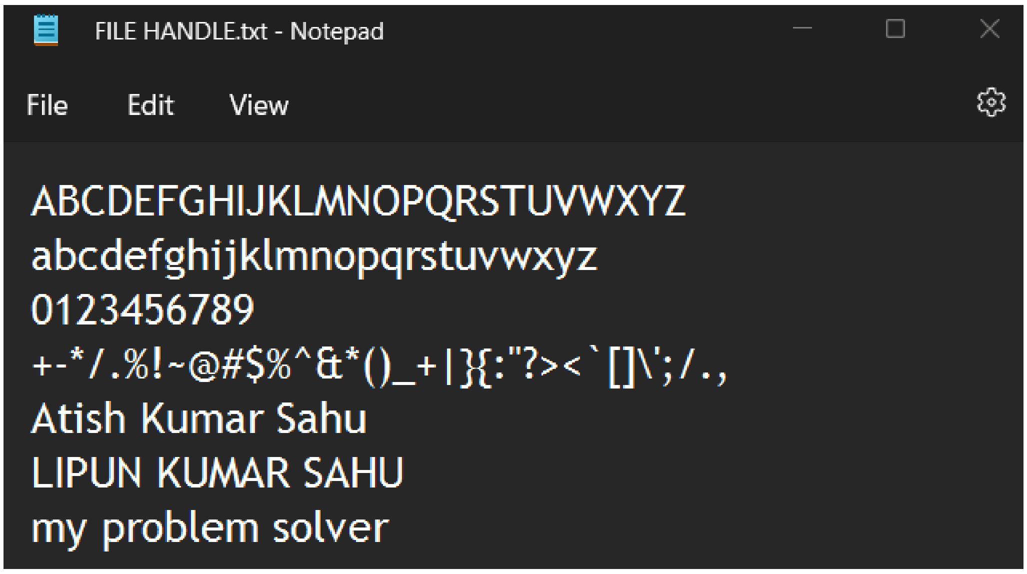


FILEH.c

```c
#include<stdio.h>

int main()
{
    FILE *fp;
    char ch;

    fp = fopen("E:\\FILE HANDLE.txt","r+");

    while(1)
    {
        ch = fgetc(fp);

        if(ch == EOF)
        break;
        printf("%c",ch);
    }
    fclose(fp);
}
```

E:\C CODE\FILEH.exe

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
+-*/.%!~@#$%^&*()_+|}{:"?><`[]\';/.,
Atish Kumar Sahu
LIPUN KUMAR SAHU
my problem solver


--------------------------------
Process exited after 0.09186 seconds with return value 0
Press any key to continue . . .
```



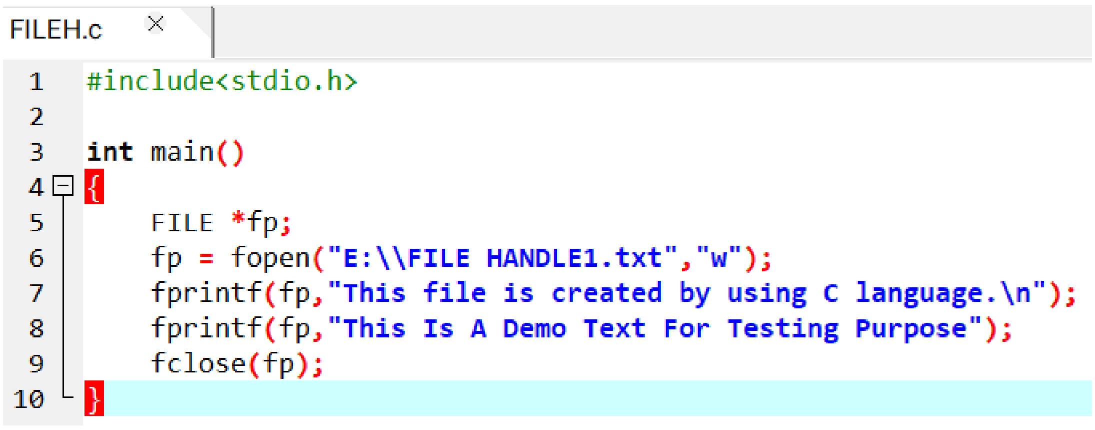
FILE HANDLE.txt - Notepad

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
+-*/.%!~@#$%^&*()_+|}{:"?><`[]\';/.,
Atish Kumar Sahu
LIPUN KUMAR SAHU
my problem solver
```

fprintf() & fscanf():

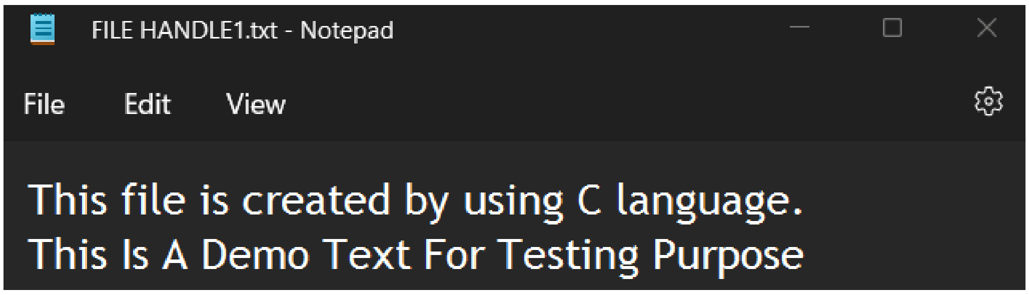
fprintf() function is used to write set of characters into file. It sends formatted output to a stream. “int fprintf(FILE *stream, const char *format [argument,...];”

fscanf() function is used to read set of characters from file. it reads a word from the file and returns EOF(End Of File) at the end of file.



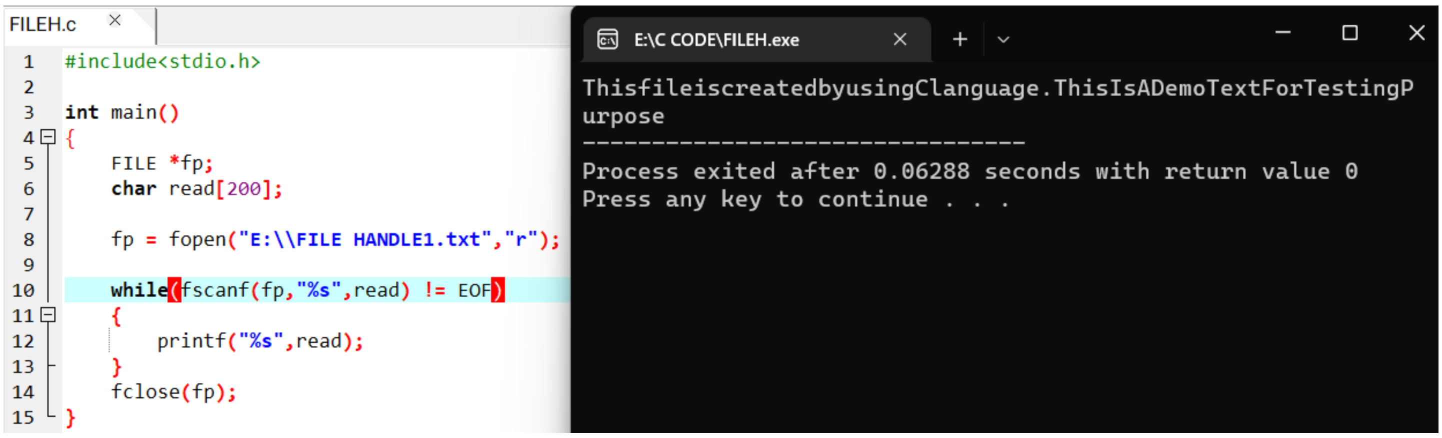
FILEH.c

```c
#include<stdio.h>

int main()
{
    FILE *fp;
    fp = fopen("E:\\FILE HANDLE1.txt","w");
    fprintf(fp,"This file is created by using C language.\n");
    fprintf(fp,"This Is A Demo Text For Testing Purpose");
    fclose(fp);
}
```



FILE HANDLE1.txt - Notepad

```
This file is created by using C language.
This Is A Demo Text For Testing Purpose
```

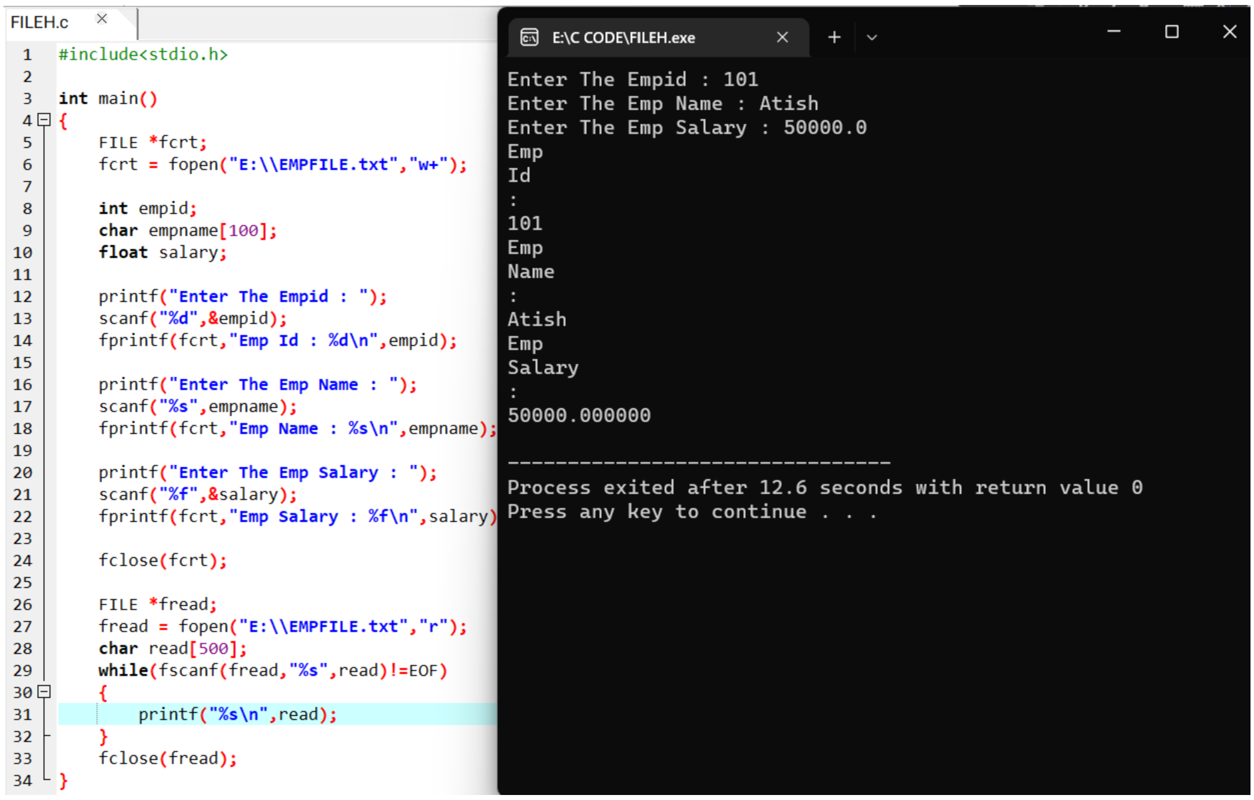


FILEH.c

```c
#include<stdio.h>

int main()
{
    FILE *fp;
    char read[200];

    fp = fopen("E:\\FILE HANDLE1.txt","r");

    while(fscanf(fp,"%s",read) != EOF)
    {
        printf("%s",read);
    }
    fclose(fp);
}
```

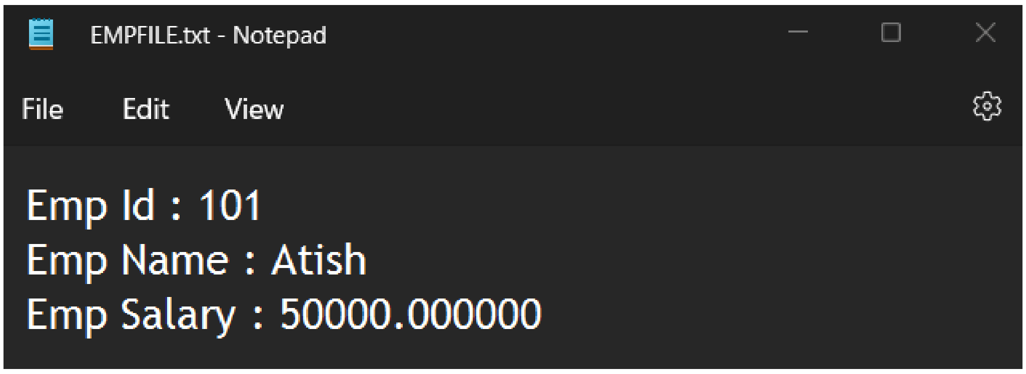
E:\C CODE\FILEH.exe

```
ThisfileiscreatedbyusingClanguage.ThisIsADemoTextForTestingP
urpose
--------------------------------
Process exited after 0.06288 seconds with return value 0
Press any key to continue . . .
```



```c
#include<stdio.h>

int main()
{
    FILE *fcrt;
    fcrt = fopen("E:\\EMPFILE.txt","w+");

    int empid;
    char empname[100];
    float salary;

    printf("Enter The Empid : ");
    scanf("%d",&empid);
    fprintf(fcrt,"Emp Id : %d\n",empid);

    printf("Enter The Emp Name : ");
    scanf("%s",empname);
    fprintf(fcrt,"Emp Name : %s\n",empname);

    printf("Enter The Emp Salary : ");
    scanf("%f",&salary);
    fprintf(fcrt,"Emp Salary : %f\n",salary)

    fclose(fcrt);

    FILE *fread;
    fread = fopen("E:\\EMPFILE.txt","r");
    char read[500];
    while(fscanf(fread,"%s",read)!=EOF)
    {
        printf("%s\n",read);
    }
    fclose(fread);
}
```

```
Enter The Empid : 101
Enter The Emp Name : Atish
Enter The Emp Salary : 50000.0
Emp
Id
:
101
Emp
Name
:
Atish
Emp
Salary
:
50000.000000

--------------------------------
Process exited after 12.6 seconds with return value 0
Press any key to continue . . .
```

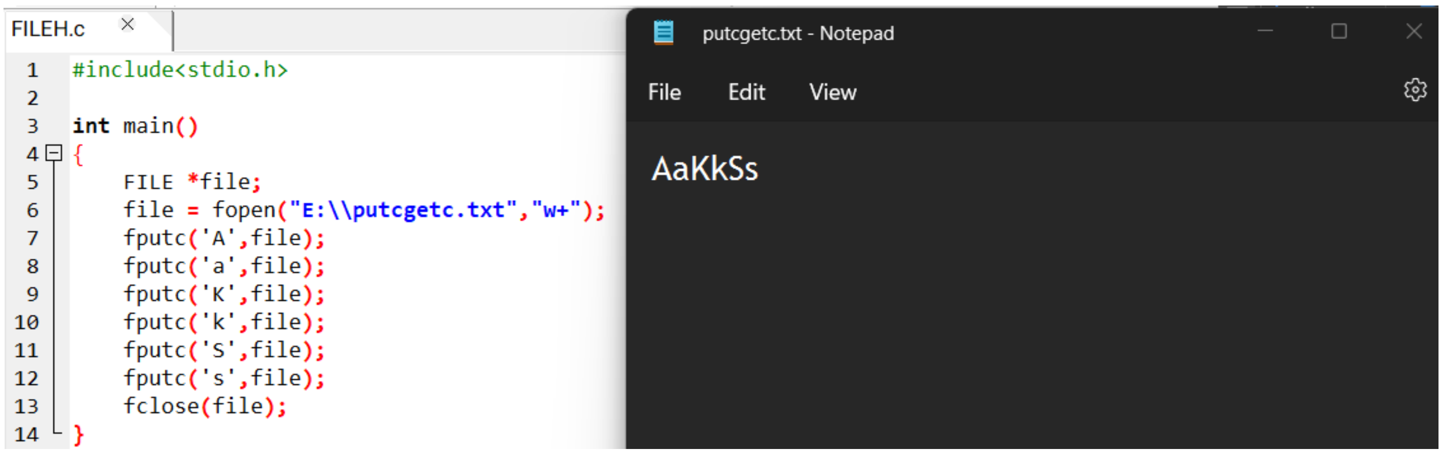


```
Emp Id : 101
Emp Name : Atish
Emp Salary : 50000.000000
```

fputc() & fgetc():

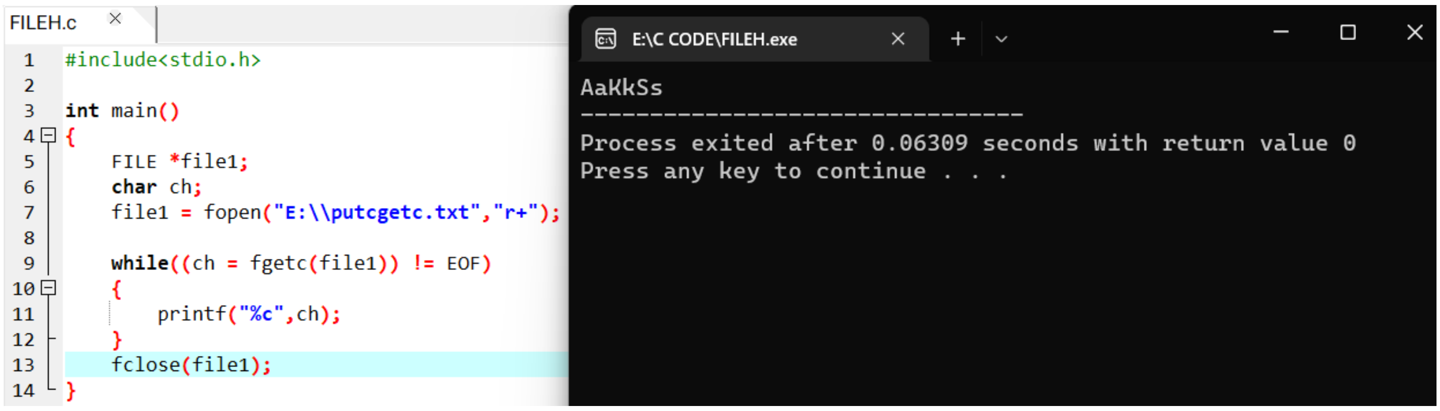
The fputc() function is used to write a single character into file. it outputs a character to a stream. “int fputc(int c, FILE *STREAM);”

The fgetc() function returns a single character from the file. it gets a character from the stream. It returns EOF at the end of the file. “int fgetc(FILE *stream);”



```c
#include<stdio.h>

int main()
{
    FILE *file;
    file = fopen("E:\\putcgetc.txt","w+");
    fputc('A',file);
    fputc('a',file);
    fputc('K',file);
    fputc('k',file);
    fputc('S',file);
    fputc('s',file);
    fclose(file);
}
```

```
AaKkSs
```


FILEH.c

```c
#include<stdio.h>

int main()
{
    FILE *file1;
    char ch;
    file1 = fopen("E:\\putcgetc.txt","r+");

    while((ch = fgetc(file1)) != EOF)
    {
        printf("%c",ch);
    }
    fclose(file1);
}
```

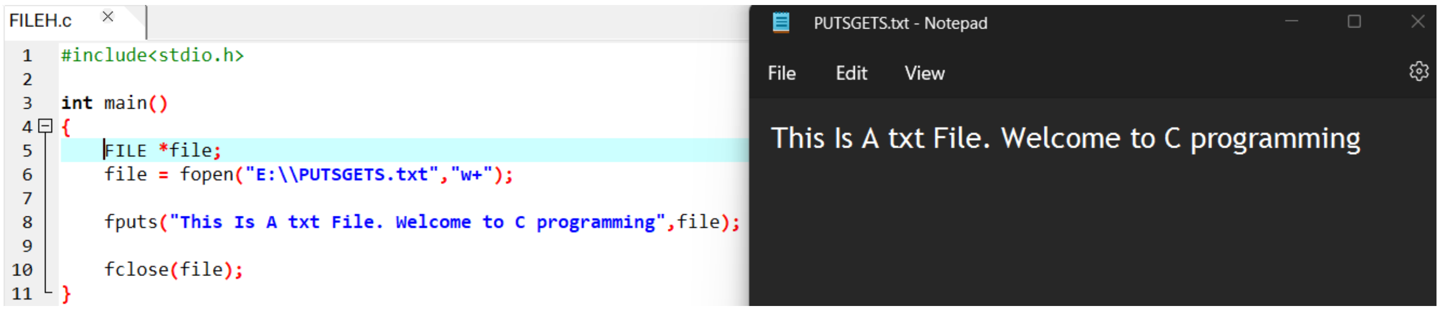
E:\C CODE\FILEH.exe

```
AaKkSs
--------------------------------
Process exited after 0.06309 seconds with return value 0
Press any key to continue . . .
```

fputs() & fgets():

The fputs() function writes a line of character into file. it outputs string to a stream. “int fputs(const char *s, FILE *stream);”

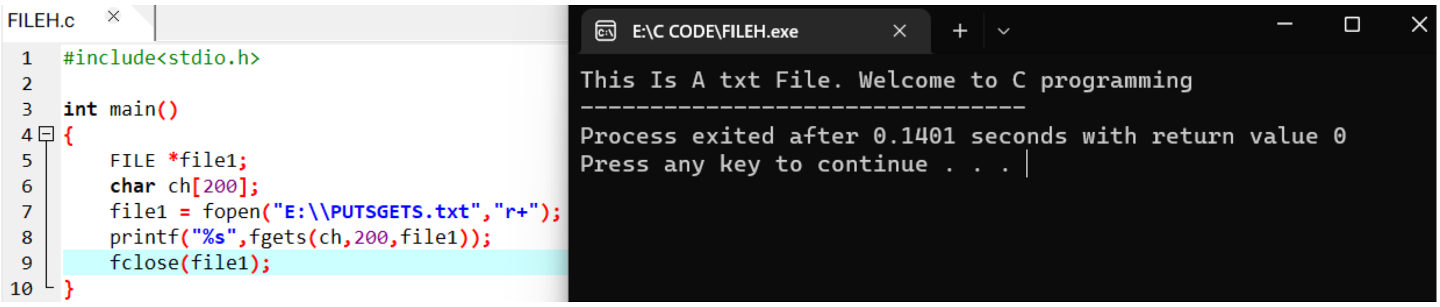
The fgets() function reads a line of character from file. It gets String from a stream. “char* fgets(char *s, int n, FILE *stream);”


FILEH.c

```c
#include<stdio.h>

int main()
{
    FILE *file;
    file = fopen("E:\\PUTSGETS.txt","w+");

    fputs("This Is A txt File. Welcome to C programming",file);

    fclose(file);
}
```

PUTSGETS.txt - Notepad

```
This Is A txt File. Welcome to C programming
```


FILEH.c

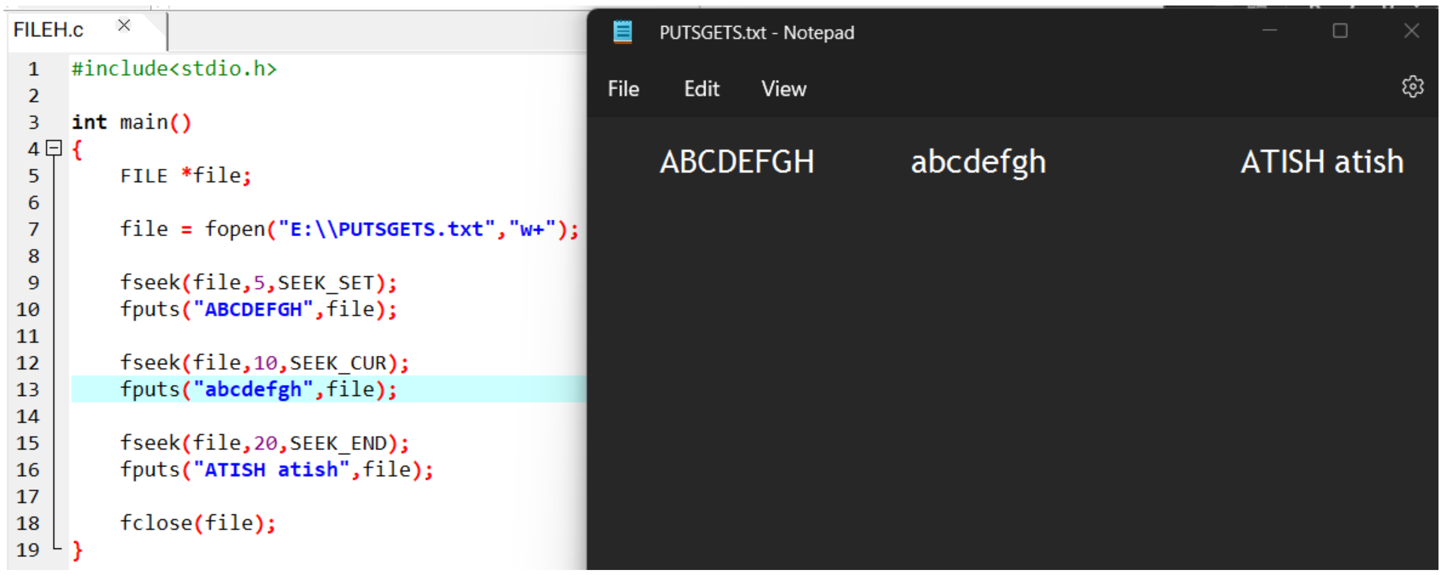
```c
#include<stdio.h>

int main()
{
    FILE *file1;
    char ch[200];
    file1 = fopen("E:\\PUTSGETS.txt","r+");
    printf("%s",fgets(ch,200,file1));
    fclose(file1);
}
```

E:\C CODE\FILEH.exe

```
This Is A txt File. Welcome to C programming
--------------------------------
Process exited after 0.1401 seconds with return value 0
Press any key to continue . . .
```

fseek() function:

The fseek() function is used to set the file pointer to the specified offset. It is used to write data into file at desired location. “int fseek(FILE *stream,long int offset, int whence);” there are 3 constants used in the fseek() function for the whence those are “SEEK_SET, SEEK_CUR, SEEK_END”.

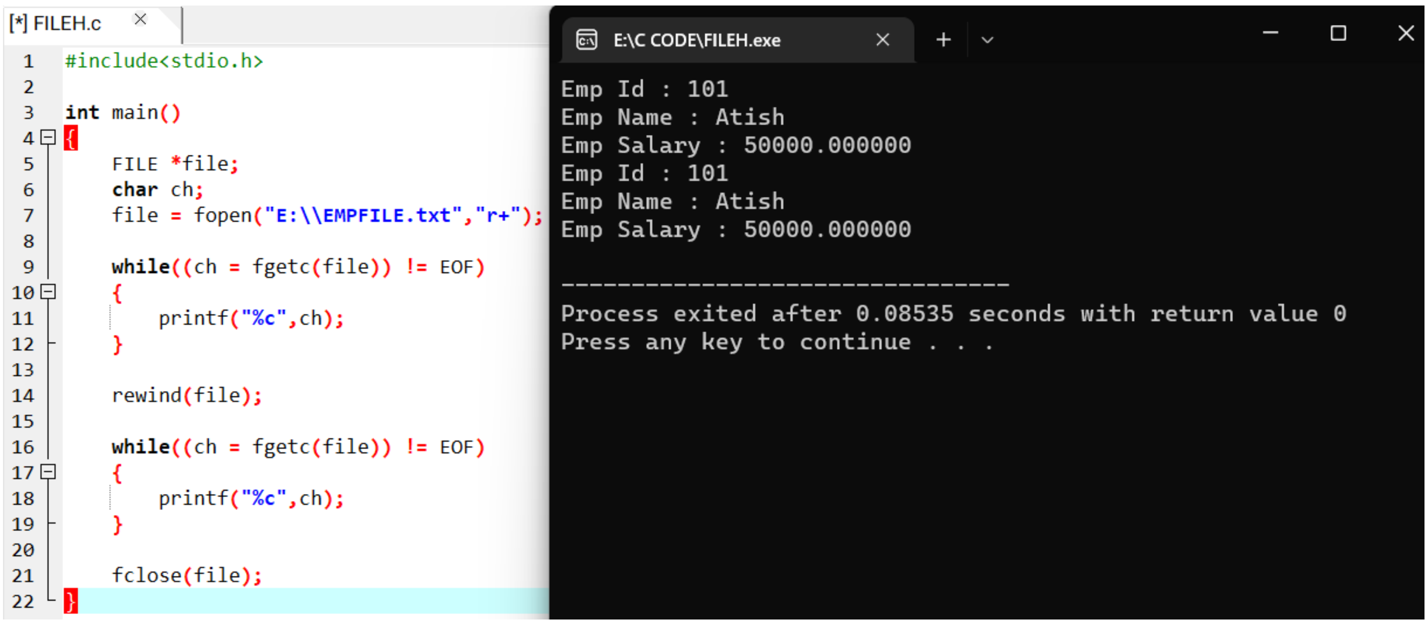


```
#include<stdio.h>

int main()
{
    FILE *file;

    file = fopen("E:\\PUTSGETS.txt","w+");

    fseek(file,5,SEEK_SET);
    fputs("ABCDEFGH",file);

    fseek(file,10,SEEK_CUR);
    fputs("abcdefgh",file);

    fseek(file,20,SEEK_END);
    fputs("ATISH atish",file);

    fclose(file);
}
```

```
     ABCDEFGH          abcdefgh                    ATISH atish
```

rewind() function:

The rewind() function sets the file pointer at beginning of the stream. it is useful if you have to stream many times. “void rewind(FILE *stream);”

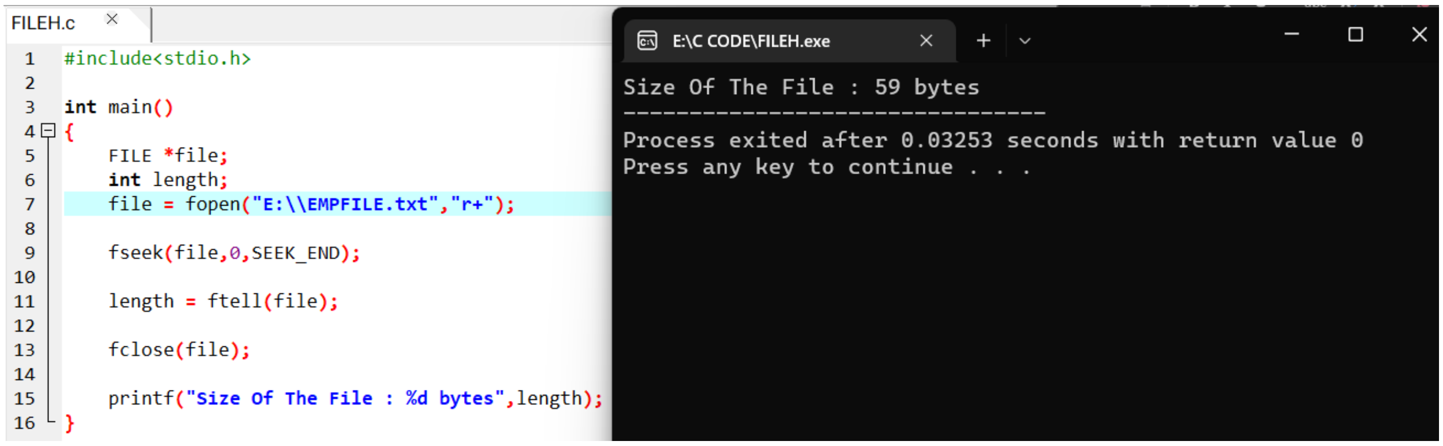


```
#include<stdio.h>

int main()
{
    FILE *file;
    char ch;
    file = fopen("E:\\EMPFILE.txt","r+");

    while((ch = fgetc(file)) != EOF)
    {
        printf("%c",ch);
    }

    rewind(file);

    while((ch = fgetc(file)) != EOF)
    {
        printf("%c",ch);
    }

    fclose(file);
}
```

```
Emp Id : 101
Emp Name : Atish
Emp Salary : 50000.000000
Emp Id : 101
Emp Name : Atish
Emp Salary : 50000.000000

--------------------------------
Process exited after 0.08535 seconds with return value 0
Press any key to continue . . .
```

ftell() function:

The ftell() function returns the current file position of the specified stream. We can use ftell() function to get the total size of a file after moving file pointer at the end of file. We can use SEEK_END constant to move the file pointer at the end of file. “long int ftell(FILE *stream);”



```c
#include<stdio.h>

int main()
{
    FILE *file;
    int length;
    file = fopen("E:\\EMPFILE.txt","r+");

    fseek(file,0,SEEK_END);

    length = ftell(file);

    fclose(file);

    printf("Size Of The File : %d bytes",length);
}
```

```
Size Of The File : 59 bytes
--------------------------------
Process exited after 0.03253 seconds with return value 0
Press any key to continue . . .
```