FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW
LUCKNOW

Computer System and Programming in ‘C’

CS-101/201

Er. Zeeshan Ali Siddiqui
Assistant Professor
Deptt. of C.S.E.

FILE HANDLING

Overview1/2

- File is a *collection* of data.
- A file is a named collection of data, stored in *secondary* storage.
- Persistent storage, not lost when machine is powered *off*.
- *Save* data in memory to files if needed (file write).
- *Read* data from file later whenever needed (file read).

Overview2/2

- It can be a *database, a program, a letter* or anything.

- 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*.

Basic Operations

- Open

- Read

- Write

- Close

- Mainly we want to do read or write, but a file has to be opened before read/write, and should be closed after all read/write is over.

Types

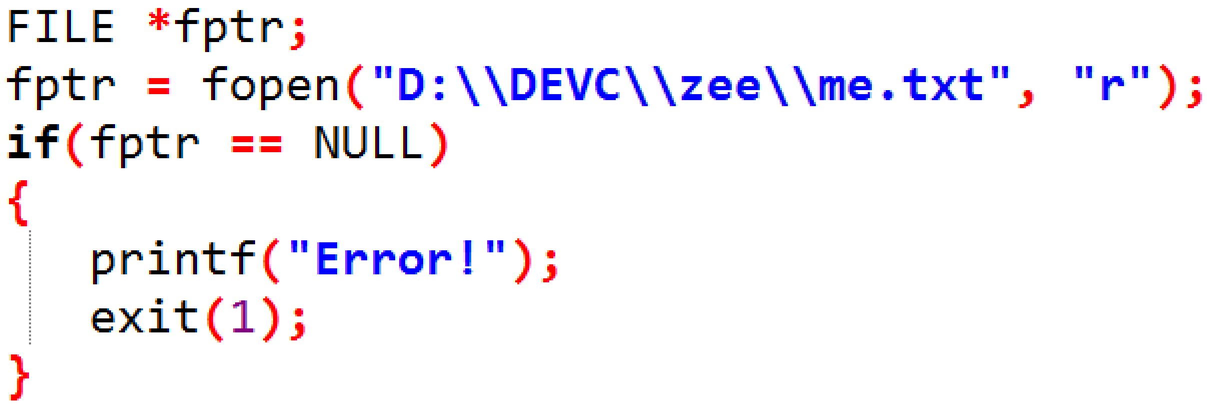
- Two kinds of files:
- Text file: contains ASCII codes only
- Binary file: can contain non-ASCII characters

Opening a File: fopen()

- FILE * is a datatype used to represent *a pointer to a file*
- *fopen* takes two parameters, the name of the file to open and the mode opened
- It returns the pointer to the file if the file is opened successfully, or it returns the special value *NULL* to indicate that it is unable to open the file.
- **Syntax:**

  fp = fopen("filename", "mode");

- **Example:**



```
FILE *fptr;
fptr = fopen("D:\\DEVC\\zee\\me.txt", "r");
if(fptr == NULL)
{
    printf("Error!");
    exit(1);
}
```

Modes$_{1/2}$

- The second argument of *fopen* is the mode in which we open the file.

➢ *"r" : opens a file for reading (can only read)*

- ✓ Error if the file does not already exists
- ✓ "r+" : allows write also

➢ *"w" : creates a file for writing (can only write)*

- ✓ Will create the file if it does not exist
- ✓ Caution: writes over all previous contents if the flle already exists
- ✓ "w+" : allows read also

➢ "a" : opens a file for appending (write at the end of the file)

- ✓ "a+" : allows read also

Modes$_{2/2}$

- We can add a “b” character to indicate that the file is a *binary* file.
  - “rb”, “wb” or “ab”

```
fptr = fopen (“foet.jpg”, “rb”);
```

exit() function

- It is used for an emergency *exit* from a program
- The exit() function, called from anywhere in your C program, will *terminate* the program at once.
- Exit is part of the *stdlib.h* library.

exit(-1);

in a function is exactly the same as

return -1;

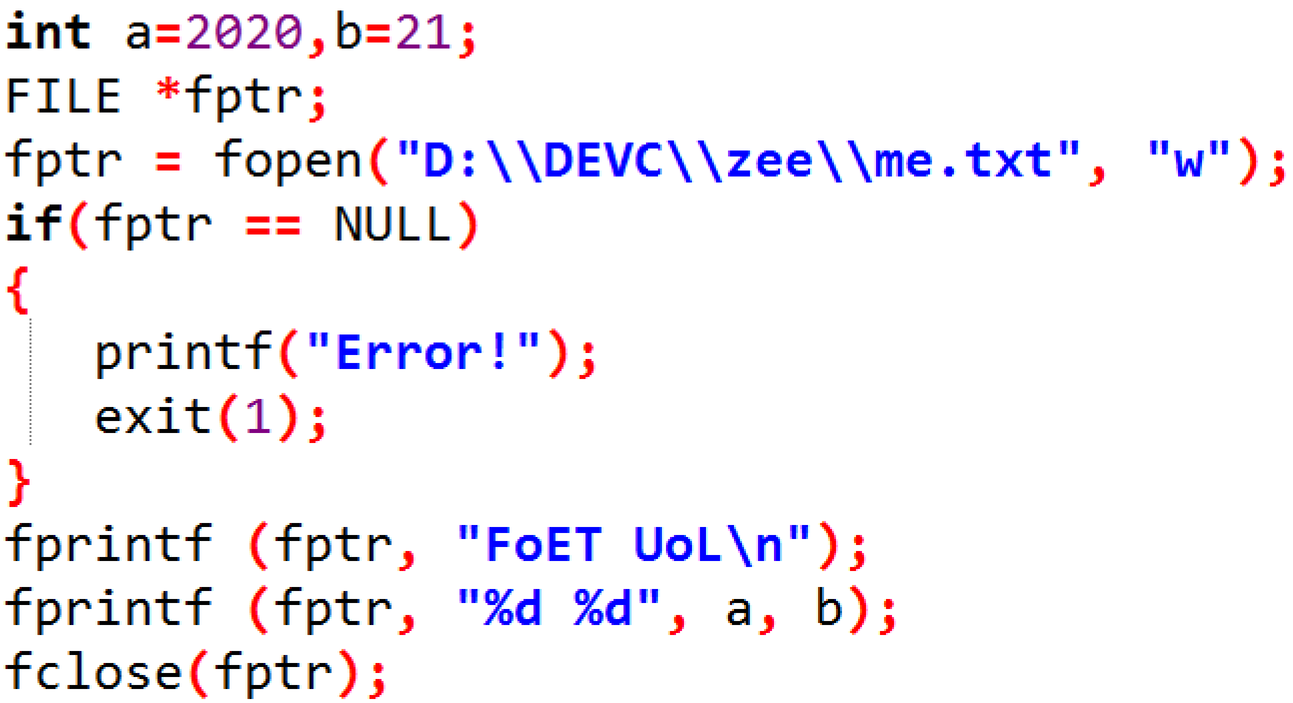
in the main function

Writing to a File: fprintf( )

- fprintf() works exactly like printf() its first argument is a file pointer. The remaining two arguments are the same a printf.
- **Syntax:**

  int fprintf(FILE *stream, const char* format, …)

- **Example:**



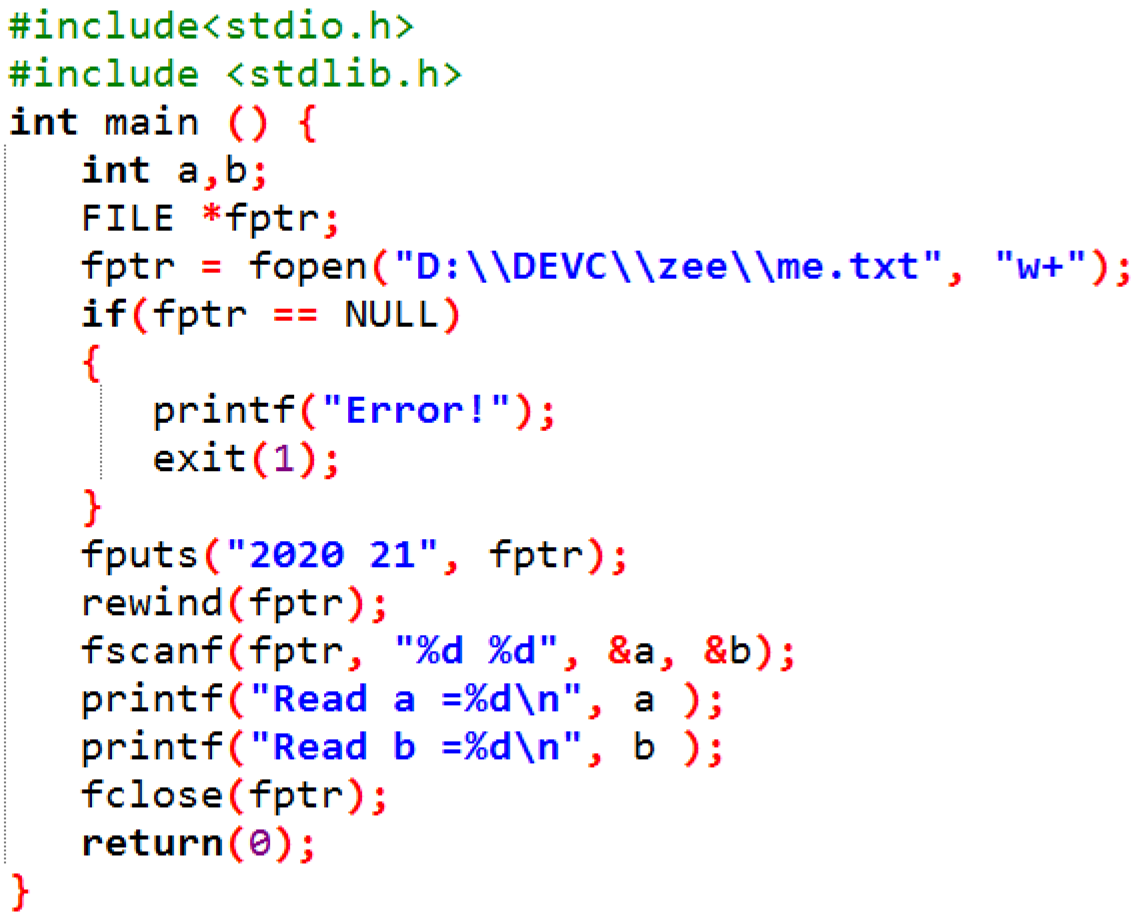
```
int a=2020,b=21;
FILE *fptr;
fptr = fopen("D:\\DEVC\\zee\\me.txt", "w");
if(fptr == NULL)
{
   printf("Error!");
   exit(1);
}
fprintf (fptr, "FoET UoL\n");
fprintf (fptr, "%d %d", a, b);
fclose(fptr);
```

Reading from a File: fscanf( )

- fscanf() works like scanf() except that its first argument is a file pointer. The remaining two arguments are the same as scanf.
- **Syntax:**

int fscanf(FILE *stream, const char* format, ...)

- **Example:**



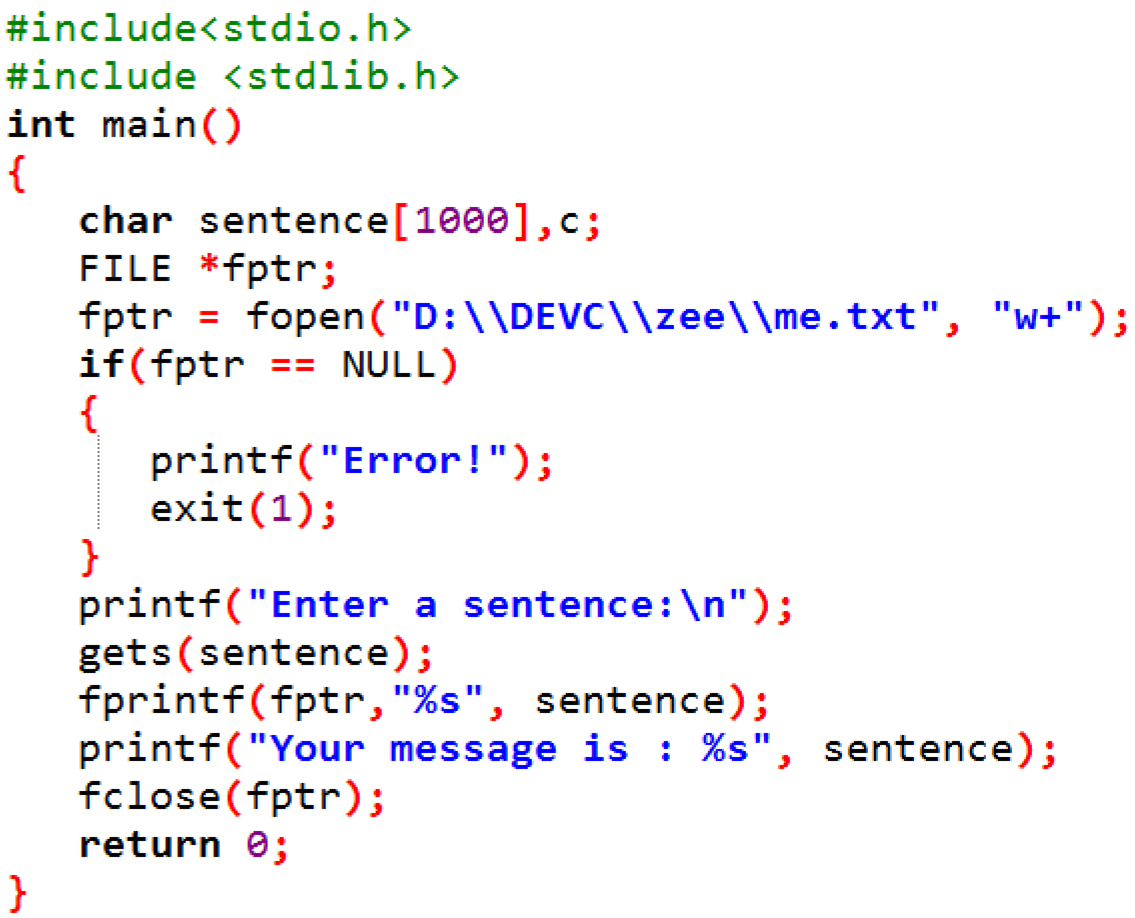
```
#include<stdio.h>
#include <stdlib.h>
int main () {
    int a,b;
    FILE *fptr;
    fptr = fopen("D:\\DEVC\\zee\\me.txt", "w+");
    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }
    fputs("2020 21", fptr);
    rewind(fptr);
    fscanf(fptr, "%d %d", &a, &b);
    printf("Read a =%d\n", a );
    printf("Read b =%d\n", b );
    fclose(fptr);
    return(0);
}
```

Closing a file: fclose()

- We should close a file when no more read/write to a file is needed in the rest of the program.
- **Syntax:**

  int fclose(FILE *file_pointer)

- **Example:**



```
#include<stdio.h>
#include <stdlib.h>
int main()
{
    char sentence[1000],c;
    FILE *fptr;
    fptr = fopen("D:\\DEVC\\zee\\me.txt", "w+");
    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }
    printf("Enter a sentence:\n");
    gets(sentence);
    fprintf(fptr,"%s", sentence);
    printf("Your message is : %s", sentence);
    fclose(fptr);
    return 0;
}
```

Homework

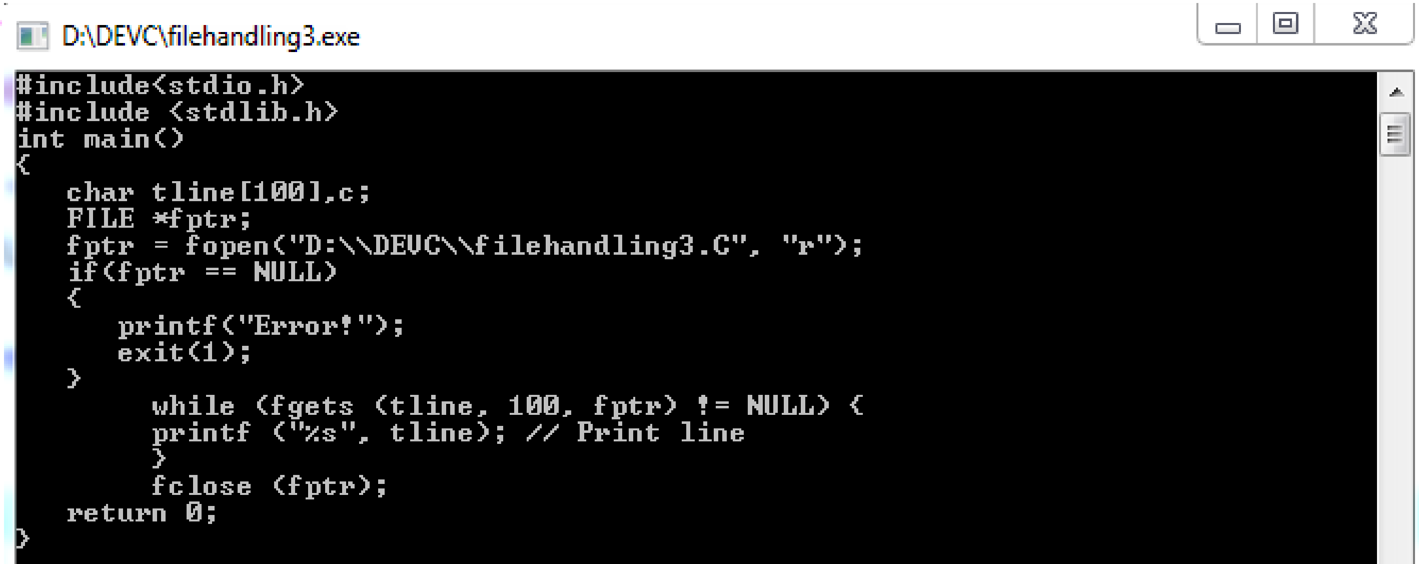
- fgetc
- fputc
- fgets

Streams in C

- Three special file streams are defined in the <stdio.h> header
- **stdin** reads input from the keyboard
- **stdout** send output to the screen
- **stderr** prints errors to an error device (usually also the screen)

Lets print a .C file on the Output Screen!

```
#include<stdio.h>
#include <stdlib.h>
int main()
{
    char tline[100],c;
    FILE *fptr;
    fptr = fopen("D:\\DEVC\\filehandling3.C", "r");
    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }
     while (fgets (tline, 100, fptr) != NULL) {
     printf ("%s", tline); // Print line
     }
     fclose (fptr);
    return 0;
}
```



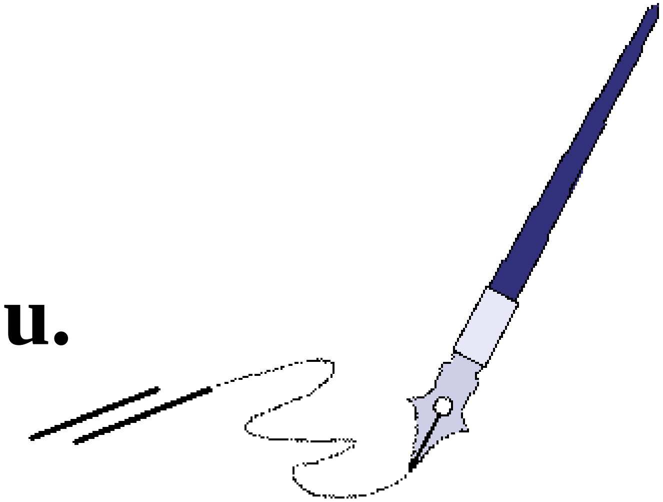
```
#include<stdio.h>
#include <stdlib.h>
int main()
{
    char tline[100],c;
    FILE *fptr;
    fptr = fopen("D:\\DEVC\\filehandling3.C", "r");
    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }
          while (fgets (tline, 100, fptr) != NULL) {
          printf ("%s", tline); // Print line
          }
          fclose (fptr);
    return 0;
}
```

Exercise

1. Define file pointer.
2. Explain different access modes used in file handling.
3. What is a file? Explain, how the file open and file close functions handled in C.
4. Write short notes on:
   - ➢ fprintf
   - ➢ fputs
   - ➢ fputc
   - ➢ Fwrite

5. Write short notes on:
   - ➢ fseek
   - ➢ ftell
   - ➢ rewind
   - ➢ fgetpos

Thank You.



BTQ

BTQ: Brain Teaser Question

There are 9 similar balls. Eight of them weigh the same and the ninth is a bit heavier.

How would you identify the heavier ball if you could use a two-pan balance scale only twice?

Given 27 table tennis balls, one is heavier than the others.

☺