



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

← [\(https://www.educba.com/static-keyword-in-c/\)](https://www.educba.com/static-keyword-in-c/)

→ [\(https://www.educba.com/queue-in-c/\)](https://www.educba.com/queue-in-c/)



Introduction to File Handling in C

File handling in C is a process where some bytes of data can be written and stored permanently in the disk so that in a later point of time, the relatable data can be fetched and referred.

Handling in C makes use of structure pointer of the file type to declare a file. For Example, An application is developed, and it is very much needed to store some important file settings then





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Most often programs are executed on terminals but in industries, the application runs should have some proof or records to be referred at some point in time. Therefore, it is very much needed to store these applications executed data somewhere and then file handling is used. It is used to write those data somewhere and save it permanently. Some pointer related structures are used to point towards that sort of file for reference. Different File handling Functions in C are as follows:

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

- **fopen [with an extra attribute such as 'a' or 'b']:** For creating a new file.
- **fopen:** Opening of an existing file.
- **fscanf or fgetc:** Reading from a file.
- **fprintf or fputs:** Writing to file.
- **rewind, fseek:** Moving to a certain or specific location within a file.
- **fclose:** Closing of a file.

Attributes to File Handling

For creating a new file using different attributes of file handling:

There are certain privileges which are needed to be provided to the file while opening or we can say kind of access control. As mentioned earlier certain specific types of pointer structures are needed for file to point these attributes are part of that only. These attributes are as follows:



- **"r":** It is used to search a file and then once the search gets complete and the file is



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

and returns null if the file is not able to open.

- **“a”**: It also works in similar fashion as that of r, but the only difference is that the pointer will point to the last character of the file. In case the file is not able to get opened it will again return a NULL value.
- **“r+”**: It is also an attribute that works the same as r just naming is different, attribute points to the first character only.
- **“w+”**: It also works the same as ‘w’ just the difference lies in the naming convention.
- **“a+”**: It also works the same as ‘a’ just the difference lies in the naming convention.

Syntax:

```
FILE *filePointer;
```

So, the file can be opened as

```
filePointer = fopen ("file.txt", "a")
```

Note: Parameters can be changed according to the above-mentioned list of attributes.

Some main functions with its syntaxes to perform some common operations are as follows:

- Reading from a file.
- Writing a file.
- Closing a file.





(<https://www.educba.com/software-development/>)

parameter, the file pointer which can be used either to read a file line by line or character by character.

Syntax:

```
FILE * filePointer;  
filePointer = fopen ("file.txt", "r");  
fscanf (filePointer, "%s %s %s %d", str1, str2, str3, &date);
```

🔗 Popular Course in this category



C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,612 ratings)

Course Price

\$79 ~~\$399~~

[View Course](https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)

Related Courses



C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software->)



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

2. Writing a file

Writing in a file can be done using both the functions `fprintf` and `fputs` in the same way as read operations.

Syntax:

```
FILE *filePointer;  
filePointer = fopen ("file.txt", "w");  
fprintf (filePointer, "%s %s %s %d", "we", "live", "in", 2020);
```

3. Closing a file

Once all operations are performed successfully and it is always asked to close the file and for closing any file it is very much needed to use `fclose` function.

Syntax:

```
FILE *filePointer;  
filePointer= fopen ("file.txt", "w");  
# Perform some file operations and then close it  
fclose(filePointer)
```

Examples to Implement File Handling in C

Below are the examples to implement in File Handling in C:



Example #1

Program for opening a file, writing and closing a file.



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
#include <string.h>

int main ()
{
    FILE *filePointer;
    char dataToWrite [50] = "Educba - portal for learning";
    filePointer = fopen ("file_handling_test.c", "w");
    if (filePointer == NULL)
    {
        printf ("file_handling_test.c file fails to get open.");
    }
    else
    {
        printf ("The file gets opened.\n");
        if (strlen (dataToWrite) > 0)
        {
            fputs (dataToWrite, filePointer);
            fputs ("\n", filePointer);
        }
        fclose(filePointer);
        printf ("Data gets successfully written in file
        file_handling_test.c\n");
        printf ("File now gets closed.");
    }
    return 0;
}
```



Output: For the main file, the output is as.



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Output of file_handling_test is as.

Example #2

Program to Open a file, Read from it and close that file.

Code:

```
#include <stdio.h>
#include <string.h>
int main ()
{
    FILE *filePointer;
    char dataToRead [50];
    filePointer = fopen ("File_Read_Test.c", "r");
    if (filePointer == NULL)
    {
        printf ("File_Read_Test.c file gets failed to open.");
    }
    else
    {
        printf ("The file then gets opened.\n");
        while(fgets (dataToRead, 50, filePointer) != NULL)
        {
            printf ("%s", dataToRead);
        }
    }
}
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
printf ("The file then gets closed.");  
}  
return 0;  
}
```

Output:

Note: A common structure is followed syntactically in terms of the file handling operations like opening a file, writing a file, reading a file and then closing a file with a mere difference of functions to be used with attributes in all scenarios.

Conclusion

File Handling in any programming language not only in C plays a very important role especially in the industry as it will store the data in memory permanently which can be referred later at any point in time. This is a special characteristic of the file handling feature.

Recommended Articles

This is a guide to File Handling in C. Here we discuss function in file handling, different attributes with examples to implement with appropriate syntax as well. You can also go through our other related articles to learn more –

1. [C Literals \(https://www.educba.com/c-literals/\)](https://www.educba.com/c-literals/)

2. [File Handling in C++ \(https://www.educba.com/file-handling-in-c-plus-plus/\)](https://www.educba.com/file-handling-in-c-plus-plus/)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

(600+ COURSES, 50+ PROJECTS)

- ☒ 600+ Online Courses
- ☒ 50+ projects
- ☒ 3000+ Hours
- ☒ Verifiable Certificates
- ☒ Lifetime Access

Learn More

[_ \(https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3\)](https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3)

About Us

Blog (<https://www.educba.com/blog/?source=footer>)

Who is EDUCBA? (<https://www.educba.com/about-us/?source=footer>)

Sign Up (<https://www.educba.com/software-development/signup/?source=footer>)

Corporate Training (<https://www.educba.com/corporate/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)

Certification Courses

All Courses (<https://www.educba.com/software-development/courses/?source=footer>)

Software Development Course - All in One Bundle
(<https://www.educba.com/software-development/courses/software-development-course/?source=footer>)





(<https://www.educba.com/software-development/>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

