



# SCM513: INTRODUCTION TO SCIENTIFIC COMPUTING PART 2: C PROGRAMMING FILE HANDLING

Elliot Menkah  
elliotsmenkah@gmail.com

November, 29, 2022





# File Handling in C

---

- Files must be opened before they can be used
- This is done by a File handler - a pointer to the file location
- All file manipulation happens through the handler
- After you are done with the file, you have to close it



<code>fopen()</code>	Opens a file. You must specify a mode
<code>fclose()</code>	Closes a file.
<code>fscanf()</code>	Scans text from file.
<code>fprintf()</code>	Prints text to file.





# File modes in C

---

"r"	Open a text file and read
"w"	Start a new text file and write to it
"a"	Open a text file and write from its ending
"r+"	Open a text file to read and write
"w+"	Create a text file to read and write
"a+"	Append or create a text file for read and write



## Example: Opening & closing a file for read

---

```
1 #include<stdio.h>
2 int main()
3 {
4     //first , declare a file pointer and initialize it to NULL
5     FILE * fPtr=NULL;
6
7     //try to open the file for reading purpose
8     fPtr=fopen("readme.txt","r");
9
10    //verify the file was opened successfully
11    if(fPtr==NULL)
12    {
13        //oops! error opening file. Terminate program
14        printf("\nFile could not be opened\n");
15        exit(1);
16    }
17
18    printf("\nFile exists and has been opened for reading\n");
19    //close file
20    fclose(fPtr);
21    printf("File closed\n");
22 }
```

## Example: Reading and printing file content

---

```
1 #include <stdio.h>
2 int main()
3 {
4     FILE * fPtr=NULL; //file pointer to access file
5     char word[50];     //string to hold words from file
6     fPtr=fopen("readme.txt","r"); //open file in read mode
7
8     //verify the file was opened successfully
9     if(fPtr==NULL)
10    {
11        //oops! error opening file. Terminate program
12        printf("\nFile could not be opened\n");
13        exit(1);
14    }
15
16    while(fscanf(fPtr,"%[^\\n]s",word)==1)
17    {
18        printf("%s\\n",word);
19    }
20    //close file
21    fclose(fPtr);
22 }
```

## Example: Opening & closing a file for write

---

```
1 #include<stdio.h>
2 int main()
3 {
4     //first , declare a file pointer and initialize it to NULL
5     FILE * fPtr=NULL;
6
7     //try to open the file for writing purpose
8     fPtr=fopen("writeme.txt","w");
9
10    //verify the file was opened successfully
11    if(fPtr==NULL)
12    {
13        //oops! error opening file. Terminate program
14        printf("\nFile could not be opened\n");
15        exit(1);
16    }
17
18    printf("\nFile has been created and ready for writing\n");
19    //close file
20    fclose(fPtr);
21    printf("File closed\n");
22 }
```

## Example: Writing to a file and saving

---

```

1 #include<stdio.h>
2 int main()
3 {
4     FILE * fPtr=NULL; //file pointer to access file
5     fPtr=fopen("writeme.txt","w"); //open file in read mode
6
7     //verify the file was opened successfully
8     if(fPtr==NULL)
9     {
10         //oops! error opening file. Terminate program
11         printf("\nFile could not be opened\n");
12         exit(1);
13     }
14
15     //Write integers from 1 to 20 to file
16     int i=0;
17     for(i=1;i<21;i++)
18     {
19         fprintf(fPtr,"%d\n",i);
20     }
21     //close file
22     fclose(fPtr);
23 }

```

Special thanks to the following for their initial work:

- *Kwesi A. Smith*
- *Shirley A. Akasreku*