# *Files handling in C++*

## REVISION OF C++ LANGUAGE

Week-10

# MODE FOR OPENING FILES

- Following syntax is used
  - FILE *fp;
  - fp = fopen ("file_name", "mode");
- A mode is used to specify whether you want to open a file for any of the below-given purposes.

| File Mode | Description |
|---|---|
| r+ | Opens a text file for both reading and writing. |
| w+ | Opens a text file for both reading and writing. It first truncates the file to zero length if it exists, otherwise creates a file if it does not exist. |
| a+ | Opens a text file for both reading and writing. It creates the file if it does not exist. The reading will start from the beginning but writing can only be appended. |

# Writing string to a file.

```
#include <iostream>
using namespace std;
int main()
{
    FILE *fpw;
    char str[100];
    fpw = fopen("Test.txt", "w");
    if (fpw== NULL)
    {
        puts("Issue in opening the Output file");
    }
    cout <<"Enter your string:";
    gets(str);
    fputs(str, fpw); //* Copied the content of str into file
    fclose(fpw);
    return 0;
}
```

Check out r + now

r + mod enables writing into the file but doesn't delete the previous contents in it.

# Writing string to a file.

```
# include<iostream>
using namespace std;
        int main ( ) {
        FILE *myfile;
        char str [100], ch;
        myfile = fopen("Test.txt", "w+");
        cout <<"\n enter a line of text:";
        gets(str);
        fputs(str,myfile);
        rewind(myfile);
while ((ch= getc (myfile)) !=EOF)
        cout<< ch;
        cout<< endl;
        fclose(myfile);
        system("pause");
        return 0;
        }
```

## *W+ & A+*

W+ mod enables writing into the file and read afterward, but you have to return the curser of the file at the beginning using rewind function.

Time to try a+ instead of w+!!

SSS.cpp*

(Global Scope)

```
#include <iostream>
# include <string>
using namespace std;
int main(){
    int a;
     FILE *fp1,*fp2;
     char s[100], st[100];
     fp1= fopen("F1.txt", "r");
     fp2= fopen("F2.txt", "r");
     if (fp1== NULL)  {
          puts("Issue in opening the Output file"); }
     while(!feof (fp1)) {
          fgets(s,100,fp1);
          fgets(st,100,fp2);
          a=strcmp(s,st);
          if (a!=0)
          cout <<"Files have not the same contents"<< endl;
     else
          cout <<"Both files have same contents"<< endl;
          fclose(fp1);
          fclose(fp2);
          system("pause");
     return 0;
}
```

100 %

# EXAMPLE-1-

write a program in C++ to compare two files and print

1- "Both files have same contents" if file1 Match file2

2- "Files have not the same contents" if file1 not Match file2

The strcmp() compares two strings character by character. If the strings are euaql, the function returns 0.

## fgets()

**Syntax :**

**char *fgets(char *str, int n, FILE *stream)**

**str:** Pointer to an array of chars where the string read is copied.

**n:** Maximum number of characters to be copied into str (including the terminating null-character).

***stream:** Pointer to a FILE object that identifies an input stream.

**returns :** the function returns str

# EXAMPLE-2-

write program in C++ to insert contents of file2 after second line of file1 and save them in file3

SSS.cpp*

(Global Scope)

```
#include <iostream>
# include <string>
using namespace std;
int main(){
    int i=0;
     FILE *fp1,*fp2, *fp3;
     char s[100], st[100];
     fp1= fopen("F1.txt", "r");
     fp2= fopen("F2.txt", "r");
     fp3= fopen("F3.txt", "w+");
         while(!feof (fp1)) {
             i++;
         fgets(s,100,fp1);
         fputs(s,fp3);
                  cout<<" i equales: "<< i<< endl;
             if(i==2) {
         while(!feof (fp2)){
         fgets(s,100,fp2);
         fputs(s,fp3); }}}
         fclose(fp1);
         fclose(fp2);
         fclose(fp3);
         system("pause");
         return 0;
}
```

# SOLVE IT FOR YOURSELF!!

Write a program in C++ that read a key from the keyboard and write it in:

1- file1 if the key is letters

2- file2 if the key is number

3- file3 the rest.

THE END