FEU INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

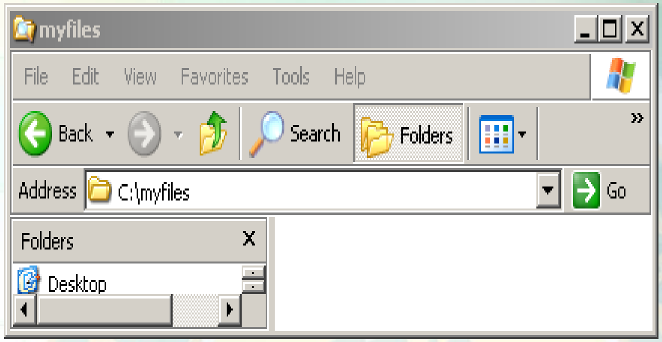
File Handling

CSPROG2

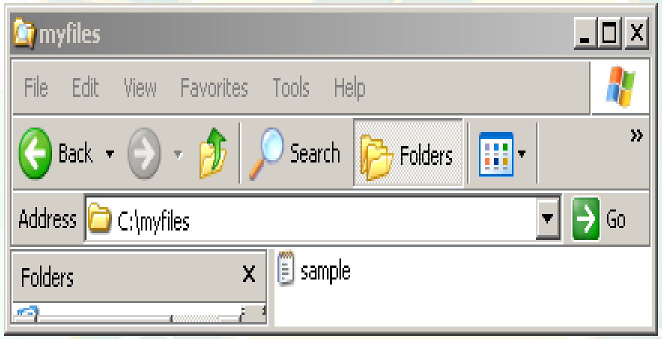
Computer Programming 2 for CS

File Handling

```
#include <stdio.h>
#include <iostream>
using namespace std;

int main()
{
        FILE *fp;
        fp = fopen("c:\\myfiles\\sample.txt","w");
        fclose(fp);
        system("pause>0");
        return 0;
}
```



Address C:\myfiles



Address C:\myfiles

sample

File Handling

- **FILE *fp;**

- File pointer points to a structure that contains about the file, such as the location of a buffer, the character position in the buffer, whether the file is being read or written, and whether errors or end of file have occurred.

- **fp=fopen("c:\\sample.txt","w");**

fopen

fopen is a library function used before reading and writing a file.

FILE *fopen(char *name, char *mode)

name – is a character string containing the name of the file.

mode – a character string, which indicates how one intends to use the file.

“r”	open text file for reading
“w”	write text file for writing; discard previous contents if any
“a”	append; open or create text file for writing at end of file
“r+”	open text file for update (i.e. reading and writing)
“w+”	create text file for update, discard previous contents if any
“a+”	append; open or create text file for update, writing at end

Reading and Writing to a File

Update mode permits reading and writing the same file

If a file that does not exist is opened for writing or appending, it is created if possible. Opening an existing file for writing cause the old contents to be discarded, while are opening for appending preserves them. Trying to read a file that does not exist is an error, and there may be other causes of error as well, like trying to read a file when you don't have permission. If there is any error, fopen will return NULL.

fclose

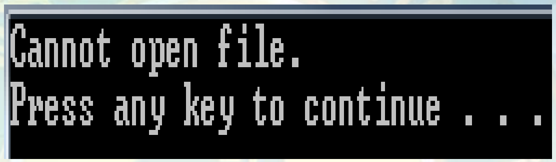
- **fclose** is the inverse of fopen, it breaks the connections between the file pointer and the external name that was established by fopen, freeing the file pointer for another file.

- **int fclose(FILE *fp);**

File Handling

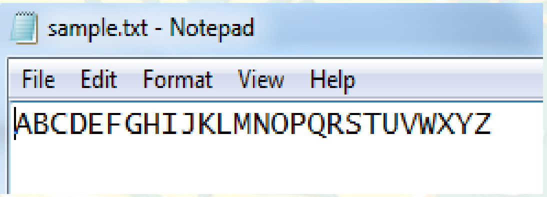
```
#include <iostream>
using namespace std;
int main()
{
        FILE *fp;
        fp=fopen("c:\\myfiles\\sample.txt","w");

        //Check permission if you can write a file
        if(!fp)
        {
                cout << "Cannot open file.\n";
                system("pause");
                exit(1);
        }
        //Prints the character ASCII value from 65 to 90 (A-Z) to a file
        for(int i=65;i<91;i++)
                fputc(i,fp);
        fclose(fp);
        return 0;
}
```

Output:
Case 1



```
Cannot open file.
Press any key to continue . . .
```

Case 2:



```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

fputc

- **fputc** write the character c to the file and returns the character written, or EOF if an error occurs.

- **int fputc(int c, FILE *fp);**

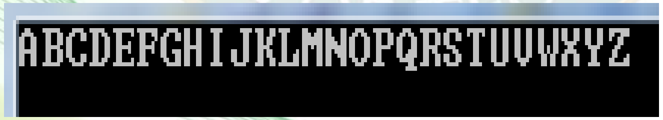
- see also **putc**

Reading from a file

```
#include <iostream>
using namespace std;
int main()
{

FILE *fp;
fp=fopen("c:\\myfiles\\sample.txt","r");

if(!fp)
        {
        cout << "Cannot open file.\n";
        system("pause");
        exit(1);
        }
```

```
//Get each character from the file and
prints to the screen
//until it reach the end of file (EOF)

        char c;
        while((c = fgetc(fp)) != EOF)
        cout << c;

        fclose(fp);
        system("pause > 0");
        return 0;

}
```

Output:



```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

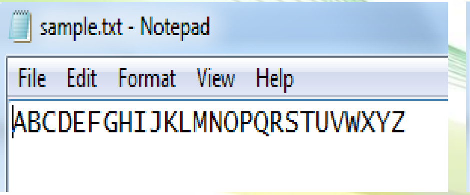
fgetc

- **fgetc** returns the next character from the stream referred to by file pointer (fp); it returns EOF for end of file or error.
- **int fgetc(FILE *fp);**
- see also **getc**
- **Appending to a file**

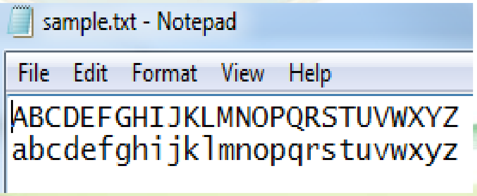
Writing to a file

```
#include <iostream>
using namespace std;
int main()
{
FILE *fp;

fp=fopen("d:\\myfiles\\sample.tx
t","a");

      if(!fp)
{
      cout << "Cannot open
file.\n";
      system("pause");
      exit(1);
}
```

```
fputc('\n',fp);

//Prints and append character
//ASCII value from 97-122 (a-z
//to a file

for(int i=97;i<123;i++)
fputc(i,fp);

fclose(fp);
return 0;
}
```



```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

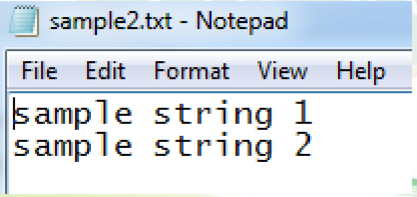


```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
```

File Line Input and Output

```
#include <iostream>
using namespace std;
int main()
{
	FILE *fp;
	fp=fopen("c:\\myfiles\\sample2.txt","w");

	if(!fp)
	{
		cout << "Cannot open file.\n";
		system("pause");
		exit(1);
	}
fputs("sample string 1\n",fp);
	fputs("sample string 2",fp);

	fclose(fp);
	return 0;
}
```

Output:



```
sample string 1
sample string 2
```

File Handling

- **fputs** writes a string (which need not contain a newline) to a file it returns EOF if an error occurs, and non negative otherwise.

- **int fputs(char *line, FILE *fp)**
-