SNS COLLEGE OF TECHNOLOGY

Coimbatore-36.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A+’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : 23CST101 PROBLEM SOLVING AND C PROGRAMMING

I YEAR/ V SEMESTER

UNIT – V STRUCTURES UNIONS AND FILES

FILES

Dr.B.Vinodhini
Associate Professor
Department of Computer Science and Engineering

UNIT V

Defining Structures and Unions– Structure declaration – Need forStructure data type-Structure within a structure -Union -Programs using structures and Unions-Pre-processor directives–Files: Opening and Closing a Data File – Reading and writing a data file – Processing a data file -Illustrative programs

FILES

A file is a collection of related data stored on a disk. C supports a wide range of functions that have the ability to perform basic **file operations**, which include:

Naming a file

Opening a file

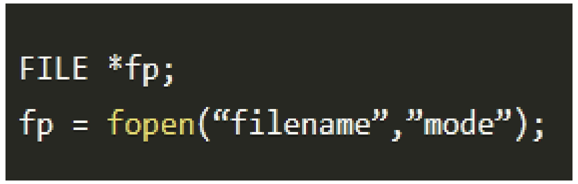
Reading data from a file

Writing data to a file

Closing a file

Features of Using Files
1.Reusablity
2.Portabilty
3.Efficient
4.Storage Capacity

Syntax for declaring and Opening File and Closing File

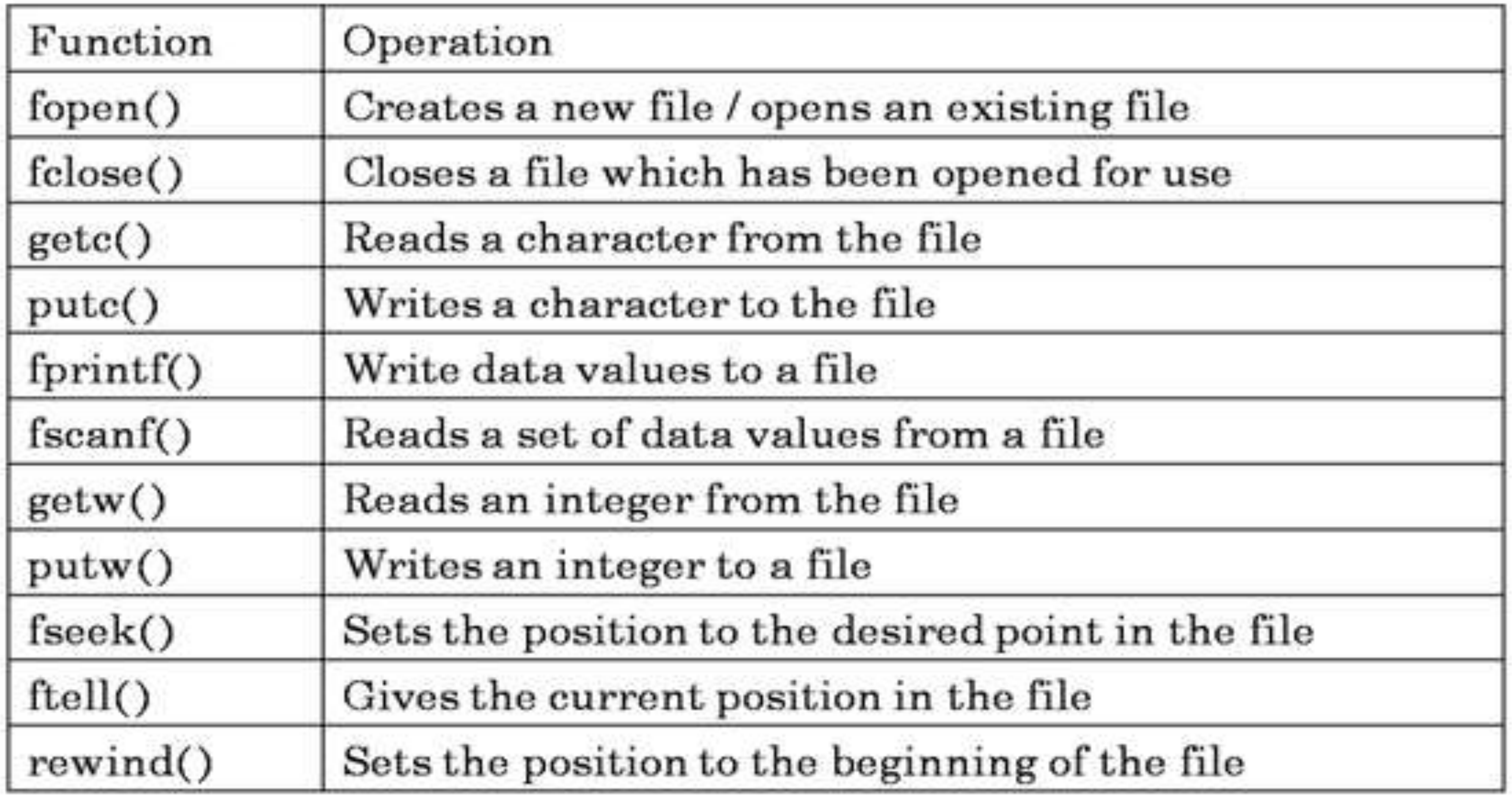


```
FILE *fp;
fp = fopen(“filename”,”mode”);
```

The **mode** specifies the purpose of opening the file. Mode can be one of the following:

r opening the file for reading data from it
w opening the file for writing data to it
a opening the file for appending data to it

FILE HANDLING FUNCTIONS



Function	Operation
fopen()	Creates a new file / opens an existing file
fclose()	Closes a file which has been opened for use
getc()	Reads a character from the file
putc()	Writes a character to the file
fprintf()	Write data values to a file
fscanf()	Reads a set of data values from a file
getw()	Reads an integer from the file
putw()	Writes an integer to a file
fseek()	Sets the position to the desired point in the file
ftell()	Gives the current position in the file
rewind()	Sets the position to the beginning of the file

FILE HANDLING FUNCTIONS

Closing the File

```
fclose(file-pointer);
```

getc and putc Functions

The simplest I/O functions are **getc** and **putc**. These are analogous to **getchar** and **putchar** functions and handle one character at a time. The **putc** function writes a character to the file associated with a file pointer. The syntax is as shown below:

```
putc(ch, file-pointer);
```

Similarly the function **getc** is used to read a character from a file associated with a file-pointer. The syntax is as shown below:

```
getc(file-pointer)
```

The file pointer moves by one character position for every operation of **getc** or **putc**. The **getc** will return an end-of-file marker EOF, when end of the file has been reached.

FILE HANDLING FUNCTIONS

getw and putw Functions

The **getw** and **putw** are integer oriented functions. They are similar to **getc** and **putc** functions and are used to read and write integers to and from files. These functions would be useful when the user is dealing with integer data. The syntax for these functions is as shown below:

```
putw(integer, file-pointer);
getw(file-pointer);
```

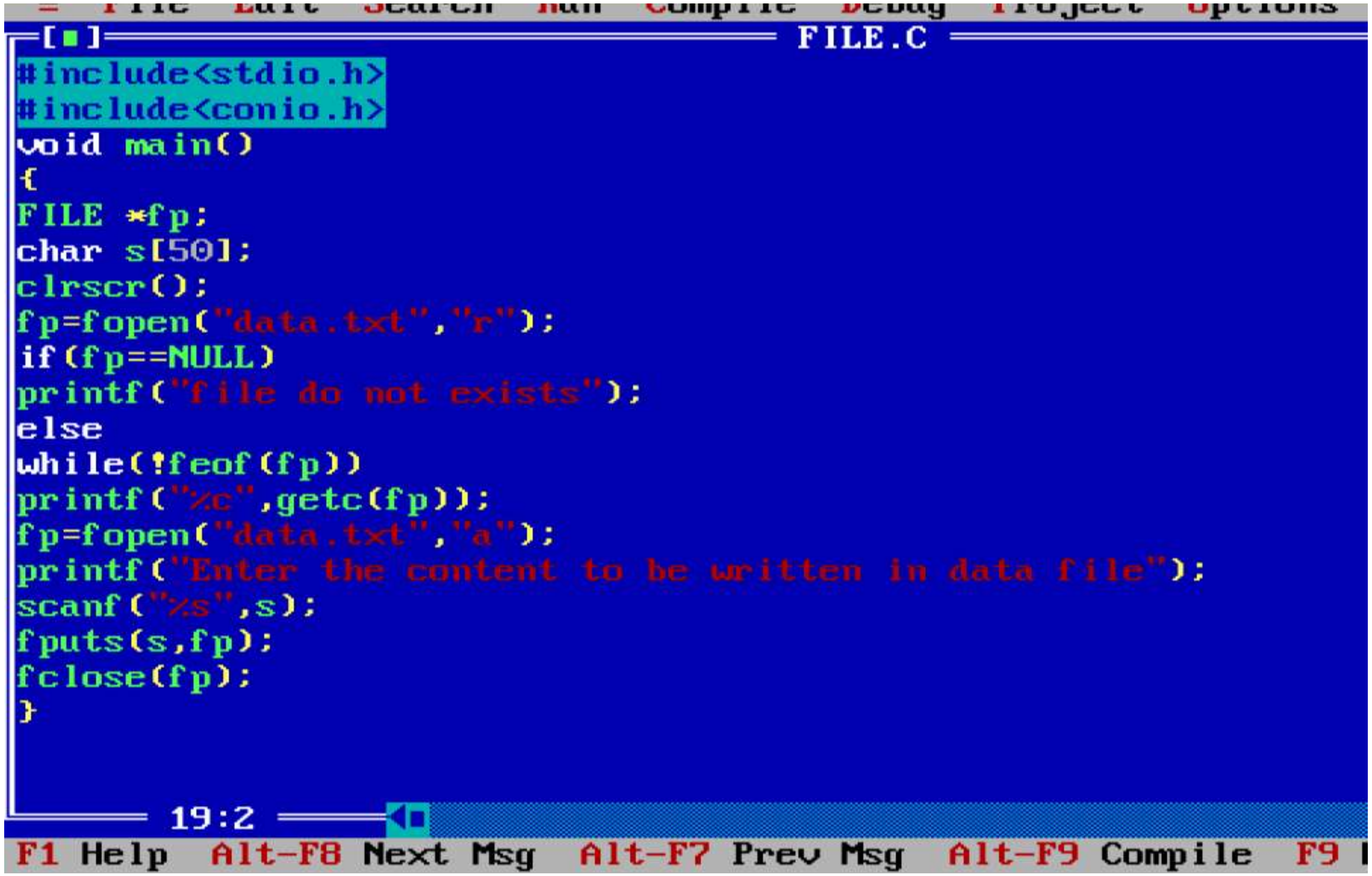
fprintf and fscanf Functions

When the user need to work with mixed data, C provides two functions namely: **fprintf** and **fscanf.** These functions are used to read and write mixed data to and from files.

These two functions are similar to **printf** and **scanf** except these two functions work on files. The syntax for these functions is as shown below:

```
fprintf(fp, “control strings”, var list);
fscanf(fp, “control strings”, var list);
```

EXAMPLE PROGRAM


[■]
FILE.C
19:2
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9

```
#include<stdio.h>
#include<conio.h>
void main()
{
FILE *fp;
char s[50];
clrscr();
fp=fopen("data.txt","r");
if(fp==NULL)
printf("file do not exists");
else
while(!feof(fp))
printf("%c",getc(fp));
fp=fopen("data.txt","a");
printf("Enter the content to be written in data file");
scanf("%s",s);
fputs(s,fp);
fclose(fp);
}
```

Thank
You