Data Files

Data Files are to store data on the memory device permanently and to access whenever is required.
There are two types of data files

1 Stream Oriented data files
2 System Oriented data files

Stream oriented data files are either text files or unformatted files. System oriented data files are more closely related to computer's operating system and more complicated to work with. In this session we go through stream oriented data files.

Opening and Closing data files

The first step is to create a buffer area where information is stored temporarily before passing to computer memory. It is done by writing

File *fp;

Here fp is the pointer variable to indicate the beginning of the buffer area and called stream pointer .
The next step is to open a data file specifying the type i.e. read only file , write only file , read /write file. This is done by using the library function fopen

The syntax is

```
fp=fopen(filename,filetype)
```

the filetype can be

1 'r' (to open an existing file for reading only)
2 'w' (to open a new file for writing only. If file with filename exists, it will be destroyed and a new file is created in its place)
3 'a' (to open an existing file for appending. If the file name does not exist a new file with that file name will be created)
4 'r+' (to open an existing file for both reading and writing)

5 'w+' (to open a new file for reading and writing. If the file exists with that name, it will be destroyed and a new one will be created with that name)

6 'a+' (to open an existing file for reading and writing. If the file does not exist a new file will be created).

For writing formatted data to a file we use the function fprintf. The syntax is

Fprintf(fp,"conversion string", value);

For example to write the name "rajan" to the file named 'st.dat'

```
File *fp;
Fp=fopen("st.dat",'w');
Fprintf(fp,"%[^\n]","rajan");
```

The last step is to close the file after the desired manipulation. This is done by the library function fclose. The syntax is

```
fclose(fp);
```

Example

1 To create a file of biodata of students with name 'st.dat'.

```
#include<stdio.h>
#include<string.h>

Tpedef struct
{
      Int day;
      Int month;
      Int year;
}date;
Typedef Struct
{
      char name(30);
      char place(30);
      int age;
      date birthdate;
}biodata;
Main()
{
      File *fp;
      biodata student;
      fp=fopen("st.dat",'w');
      Printf("Input data");
      Scanf("%[^\n]",student.name);
      Scanf("%[^\n]",student.place);
      Scanf("%d",&student.age);
      Scanf("%d",&student.birthdate.day);
      Scanf("%d",&student.birthdate.month):
      Scanf("%d",&student.birthdate.year);
      Fprintf(fp,"%s%s%d%d%d%d",student.name,student.place,student.
      age,student.birthdate.day, student.birthdate.month,
      student.birthdate.year)
      Fclose(fp);
}

Example 2:
      To write a set of numbers to a file.
#include<stdio.h>
```

```
main()
{
      file *fp;
      Int n; float x
      fp=fopen("num.dat",'w');
      Printf("Input the number of numbers");
      Scanf("%d",&n);
      For(i=1;i<=n;++i)
      {
            Scanf("%d",&x);
            Fprintf(fp,"%f\n",x);
      }
      Fclose(fp);
}
```

Processing formatted data File

To read formatted data from a file we have to follow all the various steps that discussed above. The file should be opened with read mode. To open the existing file 'st.dat' write the following syntax

file *fp;
fp=fopen("st.dat", 'r+');

For reading formatted data from a file we use the function fscanf

Example:

```
      Tpedef struct
      {
            Int day;
            Int month;
            Int year;
      }date;
      Typedef Struct
      {
            char name(30);
            char place(30);
            int age;
            date birthdate;
      }biodata;

      Main()
      {
            File *fp;
            biodata student;
            fp=fopen("st.dat",'r+');
            fscanf(fp,"%s",student.name);
            printf("%s",student.name);
            fclose(fp);
      }
```

Processing Unformatted data files

For reading and writing unformatted data to files we use the library functions fread and fwrite in the place of fscanf and fprintf.

The syntax for writing data to file 'st.dat' with stream pointer fp is
Fwrite(&student, sizeof(record),1,fp);
Here student is the structure of type biodata

Example:
To write biodata to a file

```
Tpedef struct
{
      Int day;
      Int month;
      Int year;
}date;
Typedef Struct
{
      char name(30);
      char place(30);
      int age;
      date birthdate;
}biodata;

Main()
{
      File *fp;
      fp=fopen("st.dat",'a+')
      biodata student;
      Printf("Input data");
      Scanf("%[^\n]",student.name);
      Scanf("%[^\n]",student.place);
      Scanf("%d",&student.age);
      Scanf("%d",&student.birthdate.day);
      Scanf("%d",&student.birthdate.month):
      Scanf("%d",&student.birthdate.year);
      Fwrite(&student,sizeof(record),1,fp);
      Fclose(fp);
}
```

Example 2:
To read biodata from the file.

```
Tpedef struct
{
      Int day;
      Int month;
      Int year;
}date;
Typedef Struct
{
      char name(30);
      char place(30);
      int age;
      date birthdate;
}biodata;
```

```
Main()
{
      File *fp;
      fp=fopen("st.dat",'a+')
      biodata student;
      fread(&student,sizeof(record),1,fp);
      printf("%s\n",student.name);
      printf("%s\n]",student.place);
      printf("%d\n",&student.age);
      printf("%d\n",&student.birthdate.day);
      printf("%d\n",&student.birthdate.month):
      printf("%d\n",&student.birthdate.year);
      fclose(fp);
}
```

****