FILE HANDLING

(for permanent storage of DATA)

save

1. code (program) ---- > fact.cpp

2. data ----- > file

run --> fact --> enter no 4 fact = 24

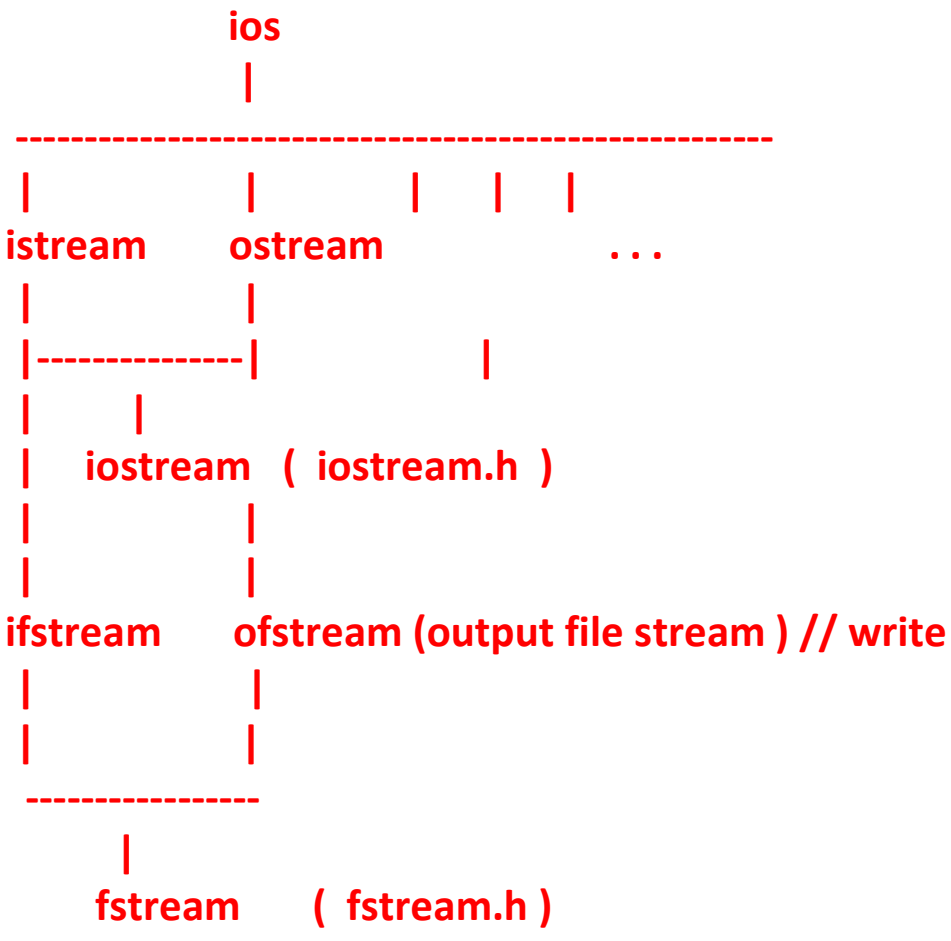
enter no 5 fact = 120

FILE OPERATION

1. **file open**
2. **file write**
3. **file read**
4. **file close**

---

1. keyboard -> input
2. file -> write
3. file -> read
4. calulation , screen -> print



FILE OPERATION

1. file open :-

a) CONSTRUCTOR METHOD

syntax

```
class_name    object(" file name ");
```

e.g. 1. ofstream f ("abc.dat"); // FILE WRITE

output file stream

2. ifstream f ("abc.dat"); // FILE READ

input file stream

b) open() :- syntax

```
object . open(" file name " , mode);
```

e.g. fstream f;

f. open("abc.dat" , ios :: in);

MODE		FILE POINTER
in	// read	start
out	// write	start
app	// append(add)	end

2) close() :- syntax

```
object . close();
```

e.g. `f . close();`

3) write() :- syntax

```
object . write( address of the record , sizeof the record );
```

e.g. `f . write( (char*) (&p) , sizeof(p));`

4) read() :- syntax

```
object . read( address of the record , sizeof the record );
```

e.g. `f. read( (char*) (&p), sizeof(p));`

BINARY FILE

```
//    FILE -> WRITE

//    INPUT   ONE  STUDENT RECORD

#include<iostream>
#include<fstream>
using namespace std;

  class student
  {

        private :  char  name[10];

                    int   roll;

       public  :   void  get()
                   {
                         cout << " ENTER NAME AND ROLL "<<endl;
                          cin  >> name >> roll;
                   }
  };

 int     main()
 {
        student   p;
```

```
        ofstream f("abc.dat"); // write

        p . get();  // keyboard -> input

        f . write( (char*) (&p) , sizeof(p) );  // file -> write

        f . close();
 }
```

```
//    FILE -> READ

                // PRINT ONE  STUDENT  RECORD

#include<iostream>
#include<fstream>
using namespace std;

  class student
  {
        private : char name[10];
             int   roll;
        public :
```

```
        void out()
        {
            cout << " name = " << name << endl;
            cout << " roll = "  << roll << endl;

                }
        };
int     main()
    {
        student  p;

        ifstream  f ("abc.dat");  //  read

        f . read( (char*) (&p) , sizeof(p) );  // file -> read

        p . out() ;  // screen -> print
}
```

MULTIPLE RECORDS

```
//  INPUT AND PRINT N STUDENTS RECORDS

#include<stdio.h> // fflush(stdin);
#include<string.h>
#include<iostream>
#include<fstream>
using namespace std;

  class   student
  {
        private :  char  name[10];
                   int    roll;
        public  :
                    void get()
                   {
                          cout<< " enter name and roll " << endl;
                          cin >> name >> roll;
                    }
                   void out()
                   {
                         cout << " name = " << name << endl;
                         cout << " roll = " << roll << endl;
                   }
                   int    getroll()
                   {
```

```
            return( roll );
        }
        char*  getname()    //  returning string (char*)
        {
            return(name);
        }
};      // student.h

int    main()
{
        char ch ;
        student  p ;

        fstream  f ;

        f. open("abc.dat", ios :: out );  // write

       do
       {
            p. get();    //    keyboard   -> input

            f. write( (char*) (&p), sizeof(p)); // file -> write

            cout<< " want to continue(y/n) " << endl;

            fflush( stdin );     // a = 3 , a = 5 (int )
            in>> ch ;
```

```
        } while( ch == 'y');
        f.close();  // input records
        f.open("abc.dat", ios :: in );  // read

        while( f.read((char*) (&p), sizeof(p)) ) // file -> read
        {
                p . out();    // screen -> print
        }

        f.close();
}
```

---

f . read((char*) (&p) , sizeof(p))

if record is not found then return zero(0)

otherwise non - zero (size of the record -> 12 byte)

```
// APPEND RECORDS

f.open("abc.dat", ios:: app );  // append(add)
```

// APPEND RECORDS

```
#include<stdio.h> // fflush(stdin);
#include<string.h>
#include<iostream>
#include<fstream>
using namespace std;

  class student
  {
        private : char name[10];
                  int   roll;

        public :
                  void  get()
                  {
                        cout<< " enter name and roll " << endl;

                        cin >> name >> roll;
                  }

                  void  out()
                  {
                        cout << " name = " << name << endl;

                        cout << " roll = " << roll << endl;
```

```
        }


        int   getroll()
        {
            return( roll );
        }

        char*  getname()  // returning string (char*)
        {
            return(name);
        }
};

    int main()
    {


        char ch;

        student p;

        fstream  f;

        f.open("abc.dat", ios::app );  // append(add)

        do
        {
```

```
                p. get(); // keyborad -> input

                f. write( (char*) (&p), sizeof(p)); // file -> write

                cout<< " want to continue(y / n) " << endl;

                fflush( stdin ); // a = 3 , a = 5

                cin>>

                }while( ch == f.close()://input

                f.open("abc.dat",ios::in );  //
while( f.read((char*) (&p), sizeof(p)) ) // file -> read
{
            f.close();

        }
```

SEARCH , MODIFY , DELETE

1. FIELD (NAME , ROLL)

2. RECORD NO

// linear or sequential search

```
#include ”student.h”
using  namespace std;

 int   main()
 {
      int   r ;     student   p;     fstream   f;

      f.open("abc.dat", ios :: in );   //  read

      cout<<" enter roll no. to be searched"<<endl;
      cin>>r;

      while(  f.read((char*)(&p),sizeof(p)) )
      {
            if (  p.getroll() == r )
            {                    p.out();          break;
            }
```

```
        }
        f.close();
}
     // search record --->  name

#include ”student.h”
using  namespace  std;

 int   main()
 {
        char  n[40] ;    student   p;    fstream   f;

        f.open("abc.dat", ios :: in );   // read

        cout<<"enter name to be searched"<<endl;

        cin>>n;

        while(  f.read((char*)(&p),sizeof(p)) )
        {
                if(  strcmp(p.getname( ),n) == 0 )
                {
                        p.out();          break;
                }
        }
        f.close();
}
```

RANDOM SEARCH (RECORD NUMBER)

```
#include ”student.h”
using namespace std;

int    main()
{
       int n ;   student   p;    fstream   f;

       f . open("abc.dat", ios :: in );   //  read

       out << " enter record no to be searched " << endl;

       cin >> n;

       f. seekg( (n-1)* sizeof(p) , ios :: beg);
            // n = 3  (n-1)* sizeof(p)   2 * 12 = 24

       f. read((char*) (&p), sizeof(p));

       p. out();

       f.close();
}
```

5. seekg() : get // read

seekp() : put // write

// TO MOVE PERTICULAR POSITON

syntax

object . seekg(no. of bytes , mode);

mode :- beg , end , cur

e.g. m = no. of bytes

1. f . seekg(0 , ios :: beg); // begin
2. f . seekg(m , ios :: beg);
3. f . seekg(0 , ios :: end); // end
4. f . seekg(-m , ios :: end);
5. f . seekg(m , ios :: cur);
6. f . seekg(-m , ios :: cur);