File handling in C++:

File: represents storage medium for storing data or information.

Streams: refer to sequence of bytes.

ofstream: represents the output streams and used for writing in files.

ifstream: represents the input streams and used for reading from files.

fstream: represents both input stream and output stream. used for reading from and writing in files.

Operations in file:

open(): creating a file.

read(): reading data from a file.

write(): writing new data to a file.

close(): closing a file.

File creation:

Filepointer.open("path", ios::mode);

mode:

out: writing

in: reading

app: appending

trunc: truncating

//creating/opening a file:

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

int main()
```

```
{
	fstream s;

	s.open("D:/JU/2022 1st sem/IT 2nd year/ab.txt", ios::out);
                                        // D:\\JU\\2022 1st sem\\IT 2nd year\\ab.txt

	if(!s) //s==NULL
	{
		cout<<"file creation failed: \n";
	}
	else
	{
		cout<<"new file created: \n";
		s.close();
	}
}
```

//writing to a file:

```
#include<iostream>
```

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

int main()
{
	fstream s;

	s.open("D:/JU/2022 1st sem/IT 2nd year/ab.txt",
ios::out);

	if(!s) //s==NULL
	{
		cout<<"file creation failed: \n";

	}
	else
	{
		cout<<"file opened: \n";
		s<<"Hello world: ";
		s.close();
	}
}
```

_____
_____

//reading from a file: [character by character]

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

int main()
{
	ifstream s;

	s.open("D:/JU/2022 1st sem/IT 2nd year/IT_1.txt", ios::in);

	if(!s) //s==NULL
	{
		cout<<"file creation failed: \n";

	}
	else
	{
```

```
        char ch;
        while(!s.eof())
        {
            s>>ch;
            cout<<ch;
        }

        s.close();
    }

}
```

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

int main()
{
    ofstream fout;
    string line;
```

```
fout.open("D:/JU/2022 1st sem/IT 2nd year/abc.txt");

cout<<"Enter data to the file: ";
while(fout)
{
        getline(cin,line); //take a line from standard input

        if(line=="-1")
         break;

         fout<<line<<endl;
}

fout.close();

ifstream fin;
fin.open("D:/JU/2022 1st sem/IT 2nd year/abc.txt");
while(fin)
{
        getline(fin,line);  //read a line from file
        cout<<line<<endl;
```

```
    }

    fin.close();

}
```

Special operations:

put(): it writes a single character to file.

get(): it reads a single character from a file.

tellp(): tells the current position of the put pointer.

```
    filepointer.tellp()
```

tellg(): tells the current position of the get pointer.

```
    filepointer.tellg()
```

seekp(): moves the put pointer (output) to the mentioned location.

```
    filepointer.seekp(no of bytes, reference mode)
```

seekg(): moves the get pointer (input) to the mentioned location.

filepointer.seekg(no of bytes, reference point)

3 reference points are passed:

ios::beg -> beginning from a file.

ios::cur -> current position in the file.

ios::end -> end of the file.

//implementing tellp() and seekp() operations in file:

```
#include<iostream>
#include<fstream>

using namespace std;

int main()
{
	fstream st;
	st.open("D:/JU/2022 1st sem/IT 2nd year/abcd.txt", ios::out);

	if(!st)
```

```
	{
		cout<<"file creation failed:" ;
	}

	else
	{
		cout<<"file created: \n" ;

		st<<"Hello";
		cout<<"File pointer position: "<<st.tellp()<<endl;
//5

		st.seekp(-1, ios::cur);
		cout<<"As per tellp: current File pointer position:
"<<st.tellp()<<endl;  //4

		st.seekp(2, ios::cur);
		cout<<"As per tellp: current File pointer position:
"<<st.tellp()<<endl; //6
	}

}
```

```
#include<iostream>
#include<fstream>

using namespace std;

int main()
{
	fstream st;
	st.open("D:/JU/2022 1st sem/IT 2nd year/abcd.txt", ios::out);

	if(!st)
	{
		cout<<"file creation failed:" ;
	}

	else
	{
		cout<<"file created: \n" ;
```

```
            st<<"Hello";
            cout<<"File pointer position: "<<st.tellp()<<endl;
//5

            st.seekp(-1, ios::cur);
            cout<<"As per tellp: current File pointer position:
"<<st.tellp()<<endl;  //4

          st.seekp(2, ios::cur);
            cout<<"As per tellp: current File pointer position:
"<<st.tellp()<<endl; //6

            st.close();
      }

        //reopening the file in input mode:

            st.open("D:/JU/2022 1st sem/IT 2nd year/abcd.txt",
ios::in);
            if(!st)
            {
                  cout<<"no such file: \n";
```

```
        }
        else
        {
            char ch;
            st.seekg(-5, ios::end);
            cout<<"\n\nAs per tellg: File pointer
position:"<<st.tellg()<<endl; //0

            st.seekg(2,ios::cur);
           cout<<"As per tellg(): File pointer
position:"<<st.tellg()<<endl; //2
            st.close();
        }


}
```

//after reaching -1, no file write operation can be performed:
[pointer does not move further in the negative direction:]

```
#include<iostream>
#include<fstream>
```

```
using namespace std;

int main()
{
	fstream st;
	st.open("D:/JU/2022 1st sem/IT 2nd year/abcd.txt", ios::out);

	if(!st)
	{
		cout<<"file creation failed:" ;
	}

	else
	{
		cout<<"file created: \n" ;

		st<<"Hello";
		cout<<"File pointer position: "<<st.tellp()<<endl; //5
```

```
            st.seekp(-4, ios::cur);     //st.seekp(-6, ios::cur); no
updation
            cout<<"As per tellp: current File pointer position:
"<<st.tellp()<<endl;  //0

            st<<"a";
            cout<<"As per tellp: current File pointer position:
"<<st.tellp()<<endl;  //0

            st.seekp(2, ios::cur);
            cout<<"As per tellp: current File pointer position:
"<<st.tellp()<<endl; // 2

            st.close();
        }
}
```

File pointers:

File pointers:

ifstream-->get

ofstream-->put

fstream--> inherits both get and put from iostream

get pointer: it points to the element to be read in the next input operation

put pointer: it points to the location where the element has to be written

---

write function: is used to write object or record (bytes of information)to the file. A record may be an array, class, object.

syntax:

fstream fout;

fout.write((char*)&obj, sizeof(obj));

&obj: initial byte of an object stored in memory.

sizeof(obj): size of the object represents the total number of bytes to be written from initial byte.

---

read function: is used to read object (sequence of bytes) from the file.

syntax:

fstream fin;

fin.read((char*)&obj, sizeof(obj));

&obj: initial byte of an object stored in the file.

sizeof(obj): size of the object represents the total number of bytes to be read from initial byte.

_____
_____
__

```
#include<iostream>
#include<fstream>
#include<stdio.h>

using namespace std;

class Student
{
```

```
int roll;
char name[50];
float marks;

void getData()
{
cout<<"Enter roll: ";
cin>>roll;

cout<<"\n Enter name: ";
cin>>name;

cout<<"\n Enter marks: ";
cin>>marks;

}

public:
void AddRecord()
{
fstream f;
Student st;
```

```
f.open("Student.dat", ios::app);
st.getData();

f.write((char*)&st, sizeof(st));
f.close();
}
};

int main()
{
	Student s;
	char ch='n';

	do
	{
		s.AddRecord();
		cout<<"Do you want to continue y/n? ";
		fflush(stdin);
		ch=getchar();

	} while(ch=='y'|| ch=='Y');
```

```
        cout<<"data written sucessfully: ";

}
_____
_____
_

#include<iostream>
#include<fstream>
#include<stdio.h>

using namespace std;

class Student
{

int roll;
char name[50];
float marks;

void putData()
```

```
{
cout<<"\n"<<roll<<"\t"<<name<<"\t"<<marks<<"\n";

}

public:
void display()
{
fstream f;
Student st;

f.open("D:\\JU\\2022 1st sem\\IT 2nd
year\\Programs\\Student.DAT", ios::in);

cout<<"\n Retrieved data: \n";

while(f.read((char*)&st,sizeof(st)))
{
        cout<<"hi";
        st.putData();

}
```

```
f.close();

}
};

int main()
{
	Student s;
	s.display();

}
```

How to write the data into a file and then read them:

```
#include<iostream>
#include<fstream>
#include<stdio.h>

using namespace std;
```

```
class Student
{

int roll;
char name[50];
float marks;

void getData()
{
cout<<"Enter roll: ";
cin>>roll;

cout<<"\n Enter name: ";
cin>>name;

cout<<"\n Enter marks: ";
cin>>marks;

}

void putData()
```

```
{
cout<<"\n"<<roll<<"\t"<<name<<"\t"<<marks<<"\n";

}

public:
void display()
{
fstream f;
Student st;

f.open("Student.DAT", ios::in);

cout<<"\n Retrieved data: \n";

while(f.read((char*)&st,sizeof(st)))
{
        cout<<"hi";
        st.putData();

}
```

```
f.close();
}

void AddRecord()
{
fstream f;
Student st;

f.open("Student.dat", ios::app);
st.getData();

f.write((char*)&st, sizeof(st));
f.close();
}



};

int main()
{
        Student s;
        char ch='n';
```

```
        do
        {
                s.AddRecord();
                cout<<"Do you want to continue y/n? ";
                fflush(stdin);
                ch=getchar();

        } while(ch=='y'|| ch=='Y');

        cout<<"data written sucessfully: ";

        s.display();
}
```