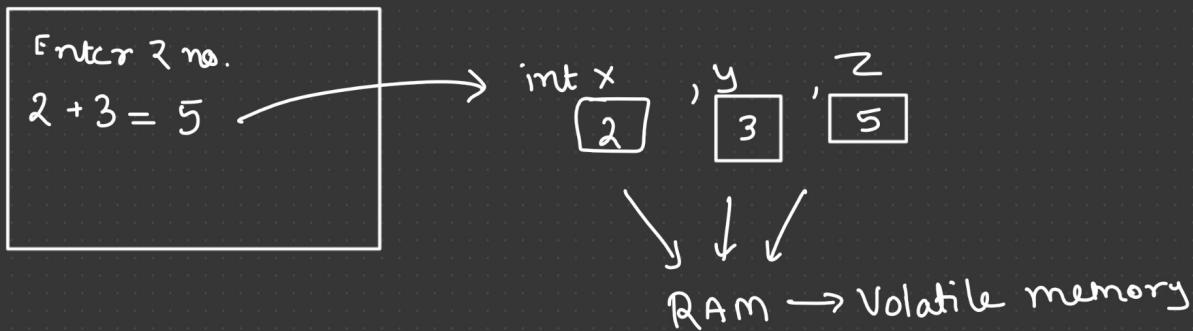


File Handling

- Storage →
- 1) Temporary
 - 2) Permanent



Permanent memory → Harddisk / Pendrive / DVD / CD / Floppy / SSD

↓ ↓ ↓

Data store + fetch (use)

→ add
→ get / update



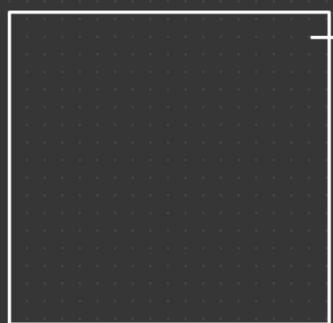
S.N	Name	Age	Add
1			
2			
3			
4			



Disadvantages of using File (Hardcopy) :-

- 1) Difficult to modify / update.
- 2) Chances of data loss.
- 3) Searching is Difficult
- 4) No option for undoing the work.
- 5) Paper wastage.
- 6) Hard to handle if data is big.

File Handling :-



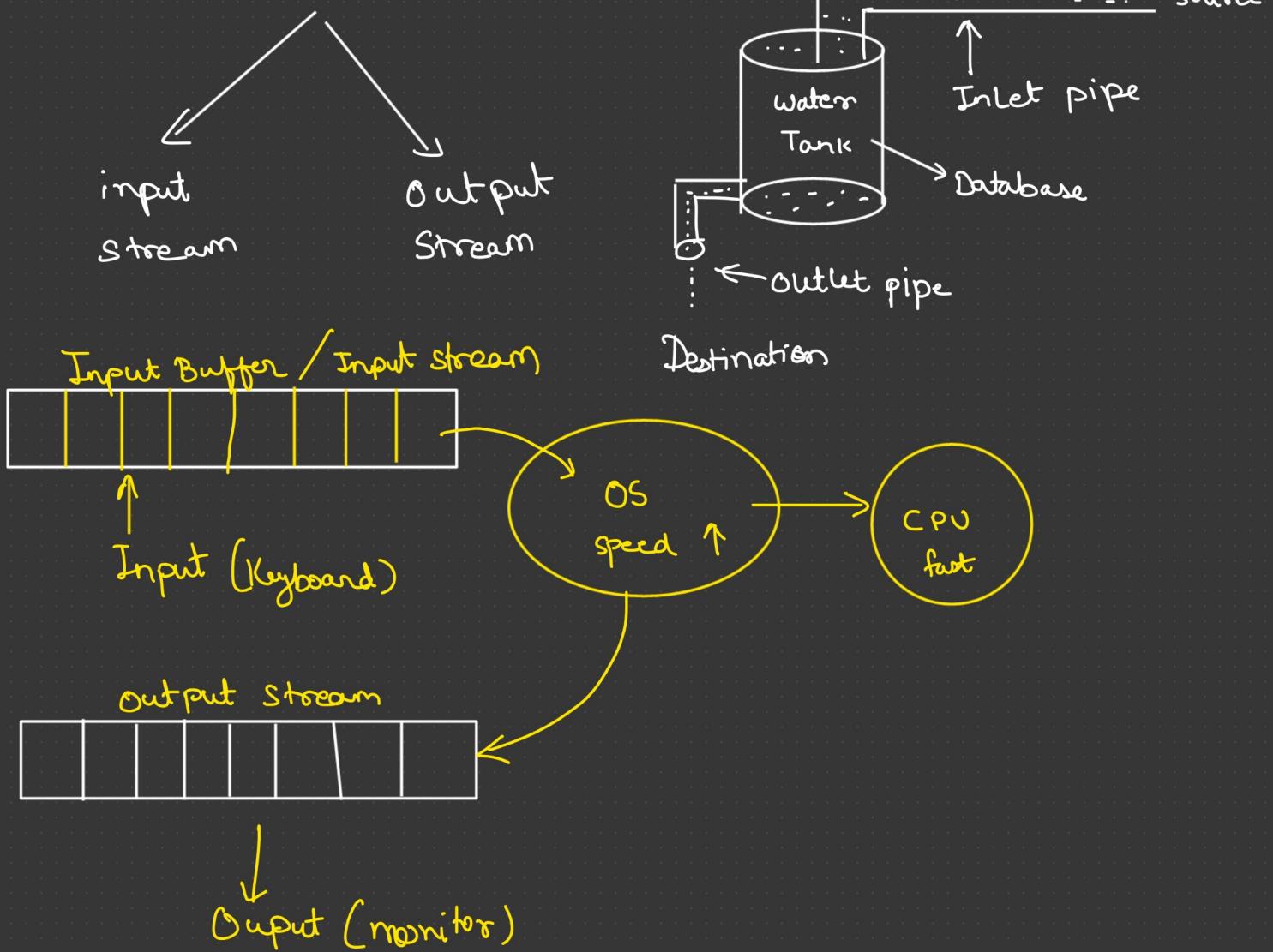
- 1) Create
- 2) Open
- 3) Write / Read
- 4) Update
- 5) Search
- 6) Delete
- 7) Close

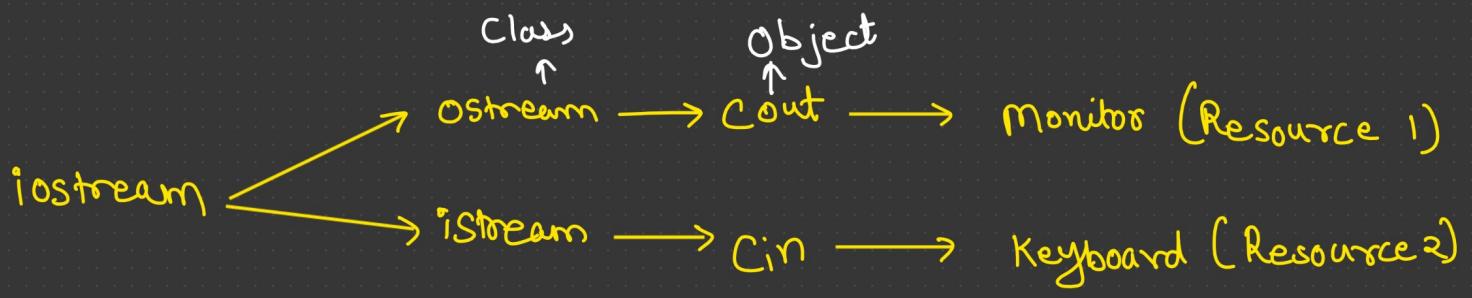
Basic functions to handle file.

In C++ mainly 3 classes are used for file Handling:-

- 1) fstream → Stream class to both read & write from/to file.
- 2) ifstream → Stream class to read from file.
- 3) ofstream → Stream class to write on file.

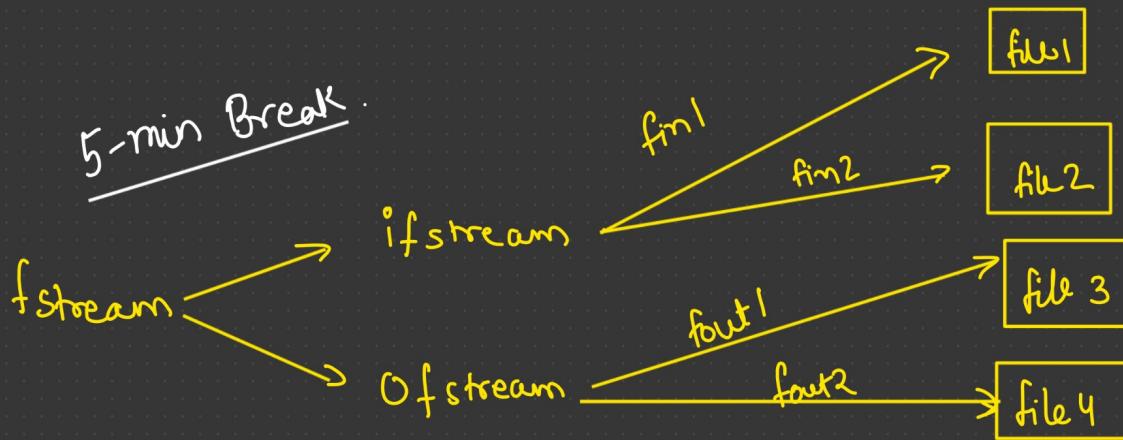
What is stream?—





What is Resource:-

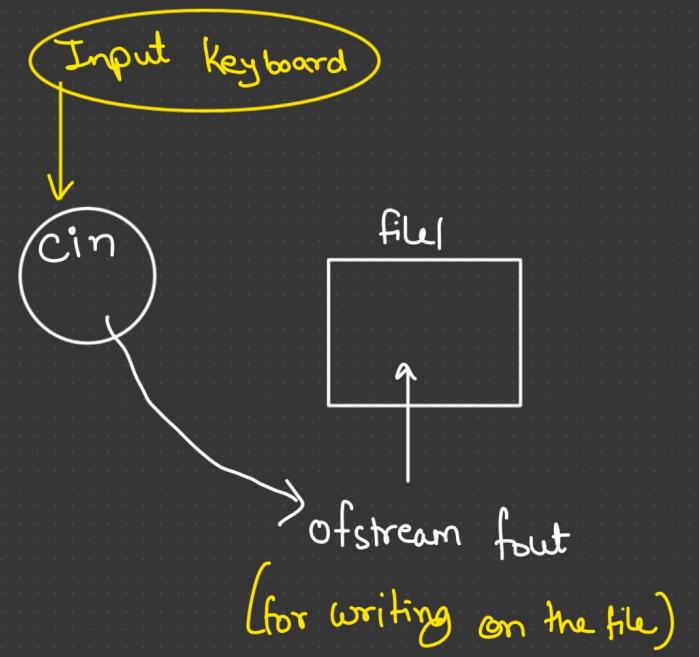
- 1) RAM
- 2) Printer
- 3) Monitor etc.



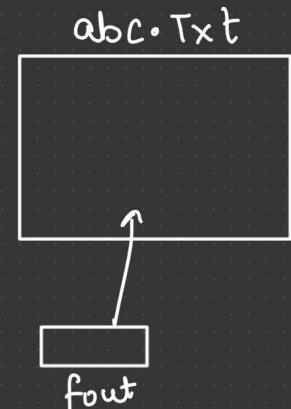
- 1) Request
- 2) Accept
- 3) Allocate
- 4) Return.

- 1) Creating a file
 - 2) Open that file
 - 3) Write operation in that file
 - 4) Closing that file

} open()



```
int main()
{
    ofstream fout;
    fout.open("C:/file/abc.txt");
    if (fout == NULL)
    {
        cout << "file not created";
    }
    else
    {
        string line;
        cout << "Enter a line";
        while (fout)
        {
            getline(cin, line);
            if (line == "-1")
                break;
            fout << line;
            fout << "\n";
        }
    }
}
```



getline

```
char a[];  
cin.getline()
```

```
String line  
getline(cin, line);
```

Open modes :-

- 1) Reading mode — `open("file name", ios::in);`
- 2) writing mode — `open("file name", ios::out);`
- 3) Truncate mode — `open("file name", ios::trunc);`
- 4) Append mode — `open("file name", ios::app);`

`open("file name", ios::in | ios::out | ios::app);`

And at last close the file using `close()` function.

`fout.close();`

Reading Data from the file:-

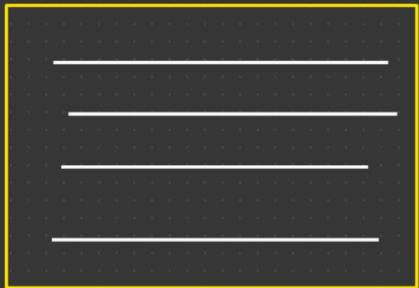
```
ifstream fin; string line;  
fin.open("C:/file/abc.Txt");  
while(!fin.eof())  
{  
    getline(fin, line);  
    cout<<line;  
}  
fin.close();
```

`eof()` → end of file
 |
 | → True → if file end
 |
 | → false → if file not ended.

1) cin (Keyboard) → cout (file write)

2) fin (file read) → cout (monitor)

Every file contains a file pointer that points to a particular location.



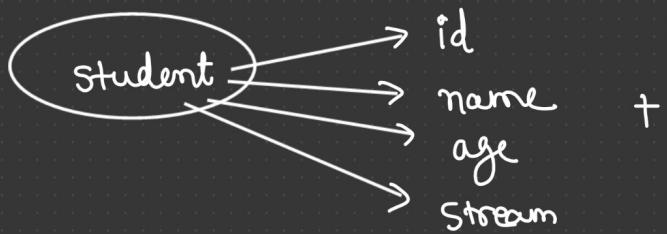
seekg() → helps to move the file pointer.

Syntax :-

fin.seekg(offset, Base address);

fin.seekg(3 , ios_base::beg);

OOPS + Database manage



+ addStudent();
print All student();
searchStudent(int id);
delete Student (int id);
update student(int id);

ID.txt

1
2

student.txt

1	}	Student S1
prateek		
23		
MCA		
2	}	Student S2
Ankit		
24		
BCA		