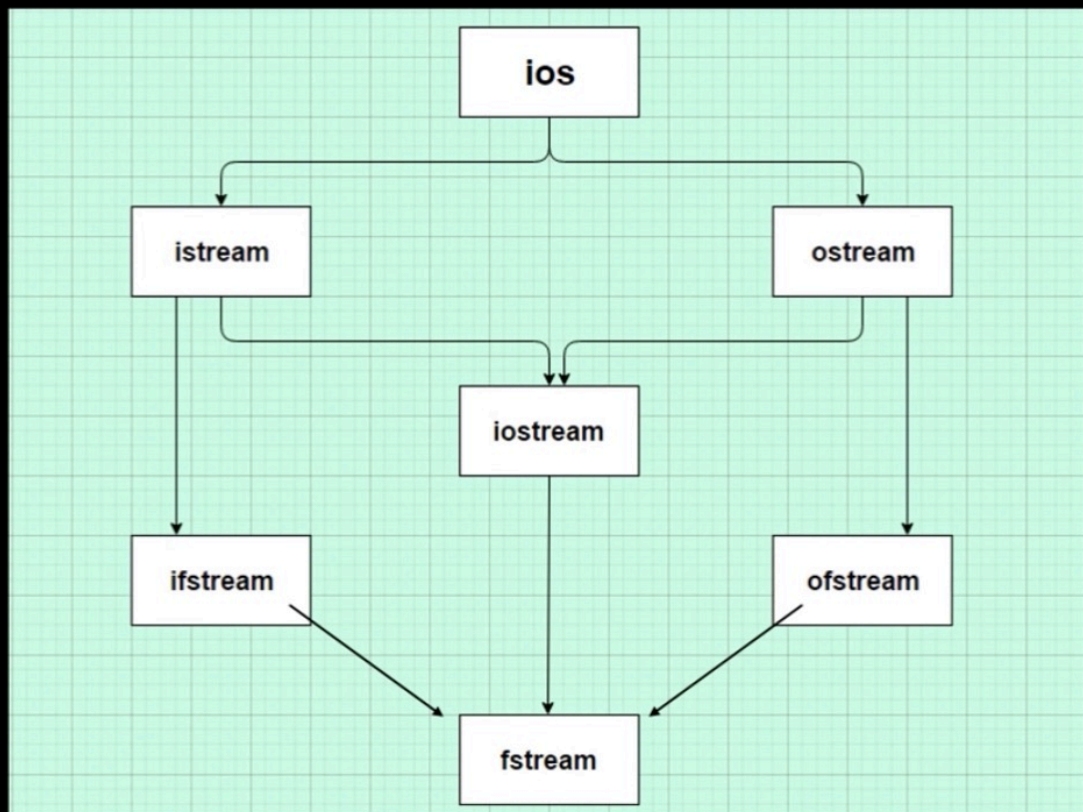


File Handling in C++

What are **files** and **streams**?

- **Files** are used to **store data permanently**.
- A **stream** is an abstraction that represents a device on which **input** and **output** operations are performed.
- A **stream** can basically be represented as a **source** or **destination** of **characters** of **indefinite length**.

File Handling in C++



File Handling in C++

| Data Type | Description |
|-----------------|---|
| ofstream | This data type represents the output file stream and is used to create files and to write information to files. |
| ifstream | This data type represents the input file stream and is used to read information from files. |
| fstream | This data type represents the file stream generally, and has the capabilities of both ofstream and ifstream which means it can create files, write information to files, and read information from files. |

File – Write Operation

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

int main()
{

char arr[100];
cout<<"Enter name & age";

cin.getline(arr,100);

ofstream myfile("xyz1.txt"); // write into file ex: int x, float y
myfile<<arr;
myfile.close();

cout<<"File write done"<<endl<<endl;

return 0;
}
```

File Read Operation

```
cout<<"Reading from file"<<endl;

char arr1[100];

ifstream obj("xyz1.txt");

obj.getline(arr1,100);

cout<<"File contents are: "<<arr1<<endl;

obj.close();
```

NOTE: above code will erase contents of files each time we execute.

If we want to store. Then use **append mode**

```
ofstream myfile("xyz1.txt",ios::app);
```

Exception handling in C++

Exception Handling in C++

- An **exception** is an **unexpected problem** that arises during the **execution** of a program.
- **Exception handling** mechanism provide a way to **transfer control** from one part of a program to another. This makes it easy to **separate** the error handling code from the code written to handle the actual functionality of the program.
- C++ exception handling is built upon three keywords: **try**, **catch**, & **throw**.

Exception Handling in C++

- **try** : A block of code which **may cause an exception** is typically placed inside the try block. It's followed **by one or more catch** blocks. If an exception occurs, it is thrown from the try block.
- **catch** : this block **catches the exception** thrown from the try block. Code to **handle** the exception is written inside this catch block.
- **throw** : A program **throws** an exception when a problem shows up. This is done using a throw keyword.
- Every **try catch** should have a **corresponding catch block**. A **single try** block can have **multiple catch** blocks.

CODE:

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

int main()
{
    try
    {
        int momsAge;
        int sonsAge;
        cout<<"enter age"<<endl;

        cin>>momsAge;
        cin>>sonsAge;

        if(sonsAge >= momsAge)
        {
            throw 99;
        }
    }
    catch(int x)
    {
        cout<<"son cannot be older than mom"<<x<<endl;
    }

    return 0;
}
```

CODE 2:

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

int main()
{
    int num,deno,result=0;
    cout<<"Enter num & deno"<<endl;
    cin>>num>>deno;
    try
    {
        if(deno==0)
        {
            throw deno;
        }
        result = num/deno; // this pt
    }

    catch(int ex)
    {
        cout<<"Divide by zero not"<<ex<<endl;
    }

    cout<<"Result is "<<result<<endl;

    return 0;
}
```

CODE 3:

```
catch(...) // default
{
    cout<<"Normal Error"<<endl;
}
```