STREAMS UNIT-5

Stream: A stream is sequence of bytes. It acts either as source from which the input data can be obtained or as a destination to which the output data can be sent. The source stream that provides data to the program is called the input stream and destination stream that receives the data is the output stream.

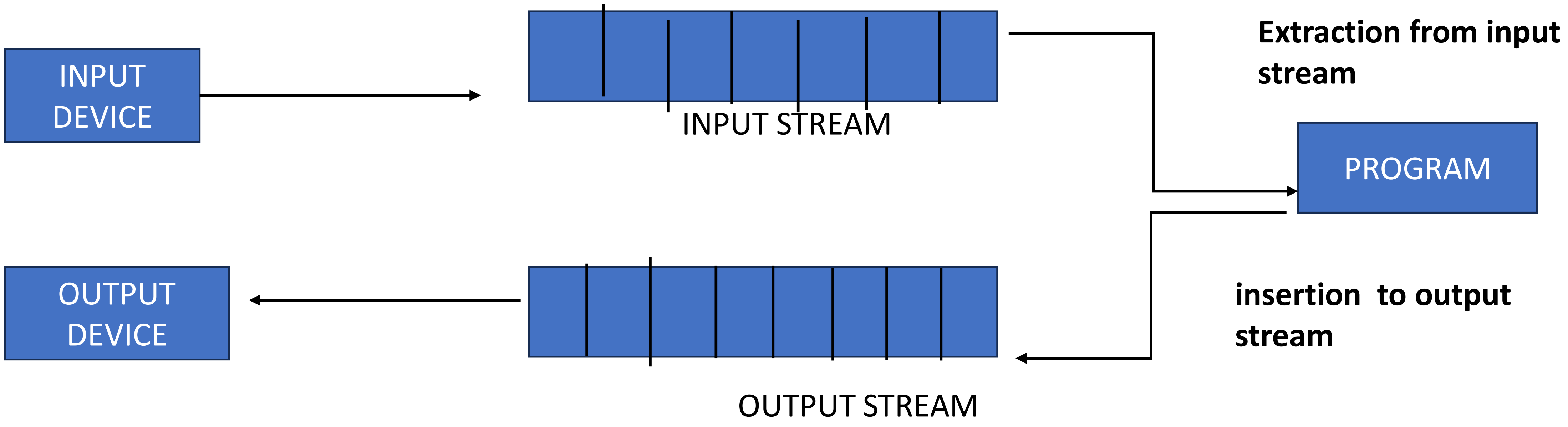
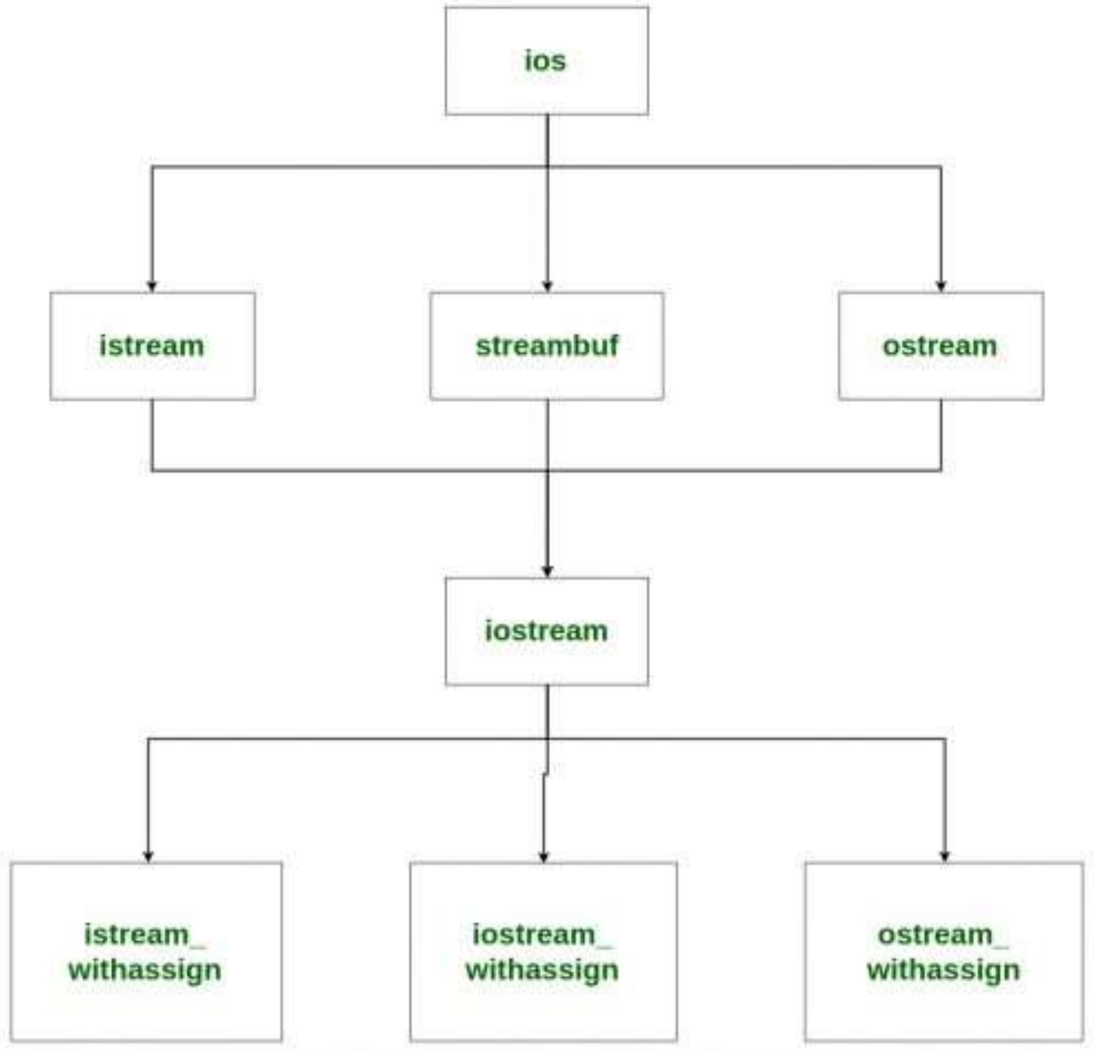


Figure of data stream

Heirarchy of Stream Classess in iostream.h



1. **ios class** is topmost class in the stream classes hierarchy. It is the base class for **istream, ostream,** and **streambuf** class.

2. **istream** and **ostream** serves the base classes for **iostream** class. The class **istream** is used for input and **ostream** for the output.

3. Class **ios** is indirectly inherited to **iostream** class using **istream** and **ostream**. To avoid the duplicity of data and member functions of **ios** class, it is declared as virtual base class when inheriting in **istream** and **ostream**

- **The ios class:** The ios class is responsible for providing all input and output facilities to all other stream classes.
- **The istream class**: This class is responsible for handling input stream. It provides number of function for handling chars, strings and objects such as get, getline, read, ignore, putback etc.
- **The ostream class**: This class is responsible for handling output stream. It provides number of function for handling chars, strings and objects such as write, put etc..
- **The iostream:** This class is responsible for handling both input and output stream as both istream class and ostream class is inherited into it. It provides function of both istream class and ostream class for handling chars, strings and objects such as get, getline, read, ignore, putback, put, write etc.
- **istream_withassign class**: This class is variant of istream that allows object assignment. The predefined object cin is an object of this class and thus may be reassigned at run time to a different istream object.

- **ostream_withassign class**: This class is variant of ostream that allows object assignment. The predefined objects cout, cerr, clog are objects of this class and thus may be reassigned at run time to a different ostream object.

Example :istream and ostream

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

int main()
{
        char x;

        // used to scan a single char
        cin.get(x);

        cout << x;
}
```

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

int main()
{
        char x;

        // used to scan a single char
        cin.get(x);

// used to put a single char onto the screen.
        cout.put(x);
}
```

- standard output stream(cout): cout is the instance of the ostream class. cout is used to produce output on the standard output device which is usually the display screen. The data needed to be displayed on the screen is inserted in the standard output stream(cout) using the insertion operator(<<).

- Standard error stream (cerr): cerr is the standard error stream which is used to output the errors. It is an instance of the ostream class. As cerr stream is un-buffered so it is used when we need to display the error message immediately and does not store the error message to display later..
- The “c” in cerr refers to “character” and ‘err’ means “error”, Hence cerr means “character error”. It is always a good practice to use cerr to display errors.

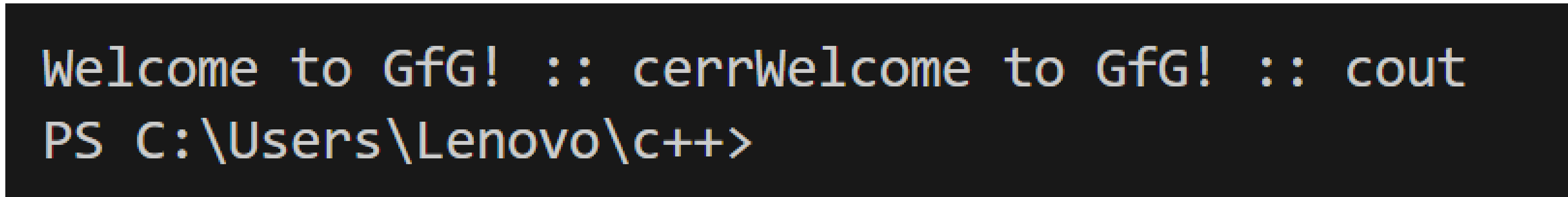
// C++ program to illustrate std::cerr

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

int main()
{

        // This will print "Welcome to GfG"
        // in the error window
        cerr << "Welcome to GfG! :: cerr";

        // This will print "Welcome to GfG"
        // in the output window
        cout << "Welcome to GfG! :: cout";
        return 0;
}
```



```
Welcome to GfG! :: cerrWelcome to GfG! :: cout
PS C:\Users\Lenovo\c++>
```

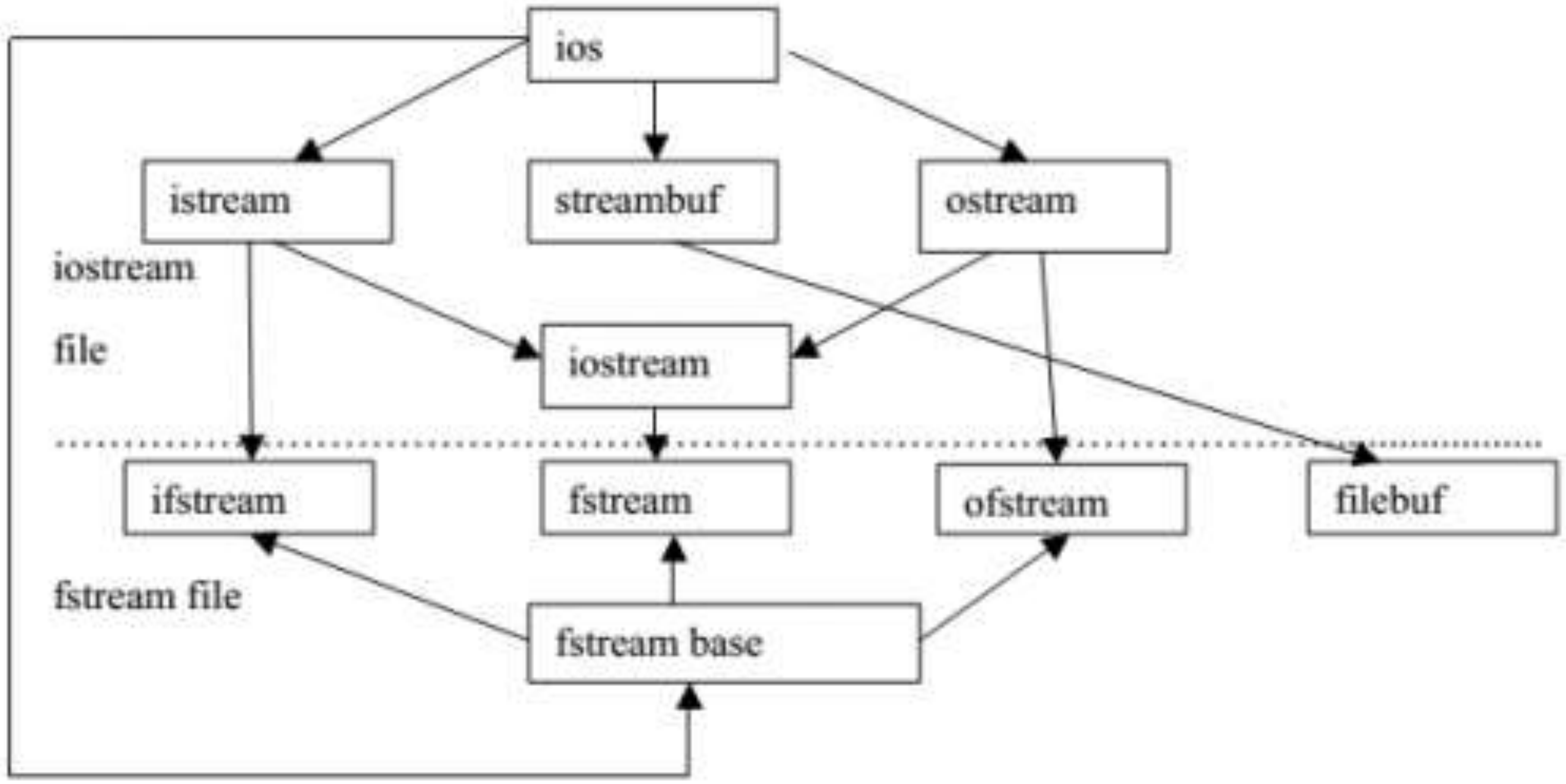


Figure 6.2 Stream classes for file operations

File handling

- File handling is used to store data permanently in a computer. Using file handling we can store our data in secondary memory (Hard disk).
- How to achieve the File Handling
- For achieving file handling we need to follow the following steps:-
- STEP 1-Naming a file
- STEP 2-Opening a file
- STEP 3-Writing data into the file
- STEP 4-Reading data from the file
- STEP 5-Closing a file

- In C++, files are mainly dealt by using three classes fstream, ifstream, ofstream available in fstream headerfile.
- ofstream: Stream class to write on files
- ifstream: Stream class to read from files
- fstream: Stream class to both read and write from/to files.

ofstream (Output File Stream)

In C++, ofstream and ifstream are classes provided by the C++ Standard Library for file I/O operations. These classes are defined in the <fstream> header and are part of the std namespace. ofstream (Output File Stream) ofstream is used to create files and write information to files.

Usage: When you want to write data to a file.

Syntax: std::ofstream outFile("filename.txt");

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

int main() {
    // Create an ofstream object
    ofstream outFile("example.txt");

    // Check if the file is open
    if (outFile.is_open()) {
        // Write to the file
        outFile << "Hello, world!" << endl;
        outFile << "This is a test." << endl;

// Close the file
        outFile.close();
        cout << "Data written to file successfully." << endl;
    } else {
        cout << "Unable to open file for writing." << endl;
    }

    return 0;
}
```

ifstream (Input File Stream) ifstream is used to read information from files.

- **Usage: When you want to read data from a file.**
- **Syntax: ifstream inFile("filename.txt");**

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

int main() {
   // Create an ifstream object
   ifstream inFile("example.txt");

   // Check if the file is open
   if (inFile.is_open()) {
      string line;
```

```
 // Read the file line by line
      while (getline(inFile, line)) {
         cout << line << endl;
      }

      // Close the file
      inFile.close();
   } else {
      cout << "Unable to open file for reading." << endl;
   }

   return 0;
}
```

Getline()

```
#include<iostream>
using namespace std;
int main(){
char name[20];
cout<<"enter ";
cin.getline(name,20);
cout<<"the name is:"<<name;
   return 0;
}
```

```
#include<iostream>

using namespace std;

int main(){
char name[20];fish
cout<<"enter ";
cin>>name;
cout<<"the name is:"<<name;
   return 0;
}
```

- The ignore function in C++ is a member function of input stream classes like istream, which includes cin, ifstream, and fstream when used for input. The ignore function is used to skip (ignore) characters in the input buffer. It is particularly useful when you want to skip unwanted characters or handle input that includes delimiters or extra characters that you don't need to process.

- **Syntax-**

istream& ignore(streamsize n = 1, int delim = EOF);

- n: The number of characters to ignore (default is 1).
- delim: The delimiter character up to which characters will be ignored (default is EOF