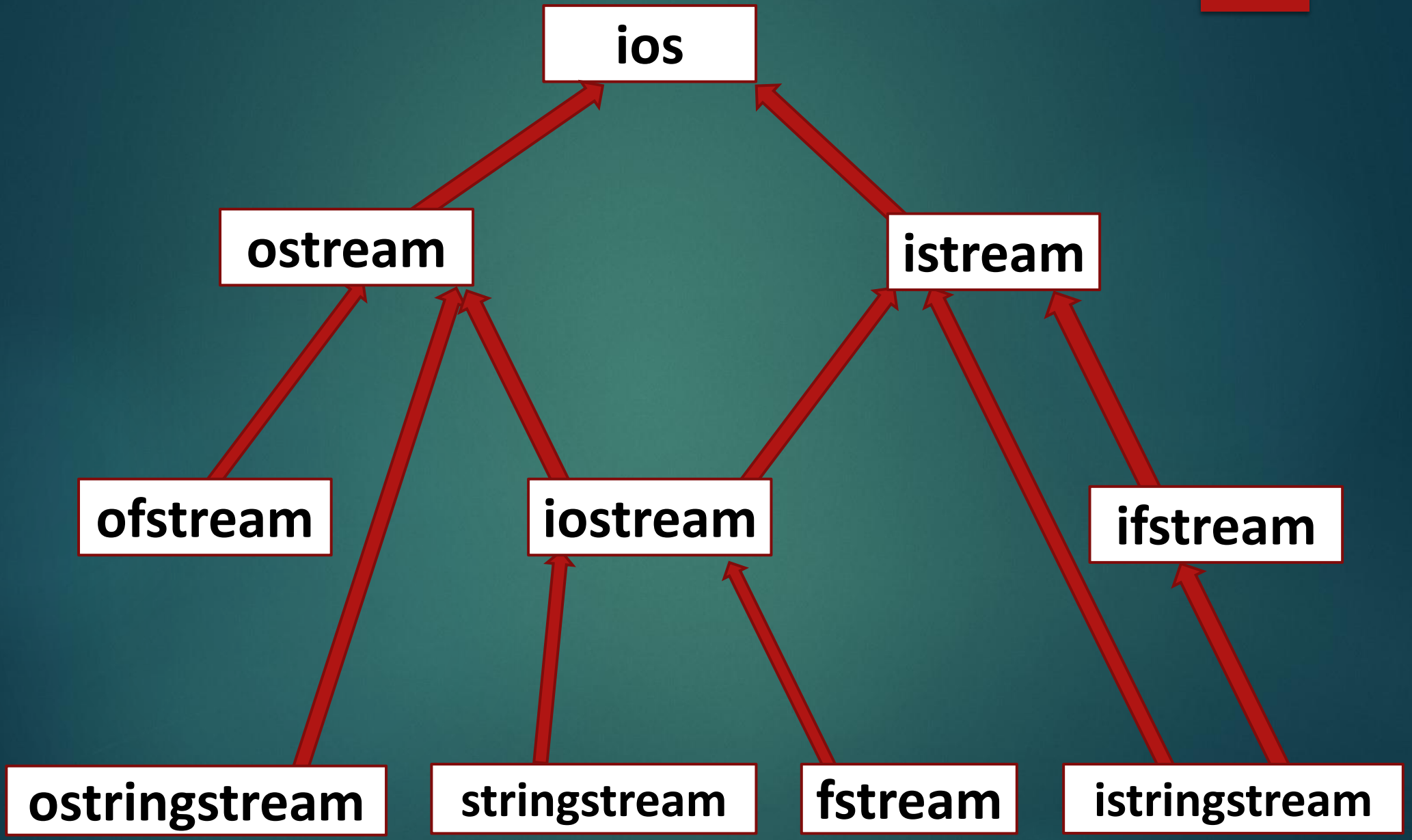
C++ Streams/File Handling

File Handling

- Text vs binary file
- C++ handles files using Streams
- C++ provides different file opening mode to perform read and write operations

C++ Streams


ios
ostream
istream
ofstream
iostream
ifstream
ostringstream
stringstream
fstream
istringstream

File Opening Modes

File Opening Modes	Meaning
ios::out	open file for output/writing. It create file if file does not exists
ios::in	open file for input/reading
ios::app	seek to end before every write/ append
ios::ate	seek to end before every immediately after opening
ios::trunc	If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one.
ios::binary	For input/output in binary format

IO Stream Library & Standard Stream Objects

- <iostream> -- basic I/O (**istream**, **ostream**, **iostream**)
- <fstream> -- file handling (**ifstream**, **ofstream**, **fstream**)
- Standard stream Objects
  - **cin** – istream class, “tied to” (connected to) the standard input device (keyboard)
  - **cout** – ostream class, “tied to” standard output device
  - **cerr** – ostream class, standard error output, unbuffered
  - **clog** – ostream class, also to standard error, buffered

Put and Get Functions

- cout.put('A'); // print a single char
- cin.get( ); // get a single char
- cin.getline(buffer, SIZE);

  // read a line of characters
- cin.eof(); // test for end-of-file

Open & Close File for Writing

- **Using scope rule**

```
int main()
{  ofstream myfile("dat.d",ios::out);
// This will open file  in out mode
    myfile <<  x;
//Insertion operator will  insert data in to file
return 0;
}
```

Open & Close File for Writing

- **Explicit open and close**

```
int main()
{
ofstream myfile;
myfile.open("dat.d", ios::out);  // This will open for writing
myfile << x;    //Insertion operator will  insert data in to file
myfile.close();
return 0;
}
```

Write in to a File


```
#include <string>
#include <fstream>
int main(){
ofstream fileobj("f.dat", ios::out);
 // create output file object
string data ="Your Name";
fileobj << data;      // output to file
return 0;
}
```

Read from File


```
#include <string>
#include <fstream>
int main(){
ifstream fileobj(“f.dat”, ios::in);
// create input file object
string data;
fileobj >> data;       // read from file
return 0;
}
```

Sequential vs Random Access of Files

- Normally, cin or cout or file stream is used sequentially
- Using the stream member functions seekg(), seekp(), read() and write( ), we can do random file access
- **Marker Positions for random access**
  - **ios::beg → Beginning of file**
  - **ios::cur → Current position**
  - **ios::end→ End of file**

Marker reposition functions

- **seekg(posistion)** → Reposition marker in input stream
- **seekp(position)** → Reposition marker in output stream
- **tellg()** → Return marker position in input stream
- **tellp()** → Return marker position in output stream

Thank You

Handle with care!!!!

……. Files