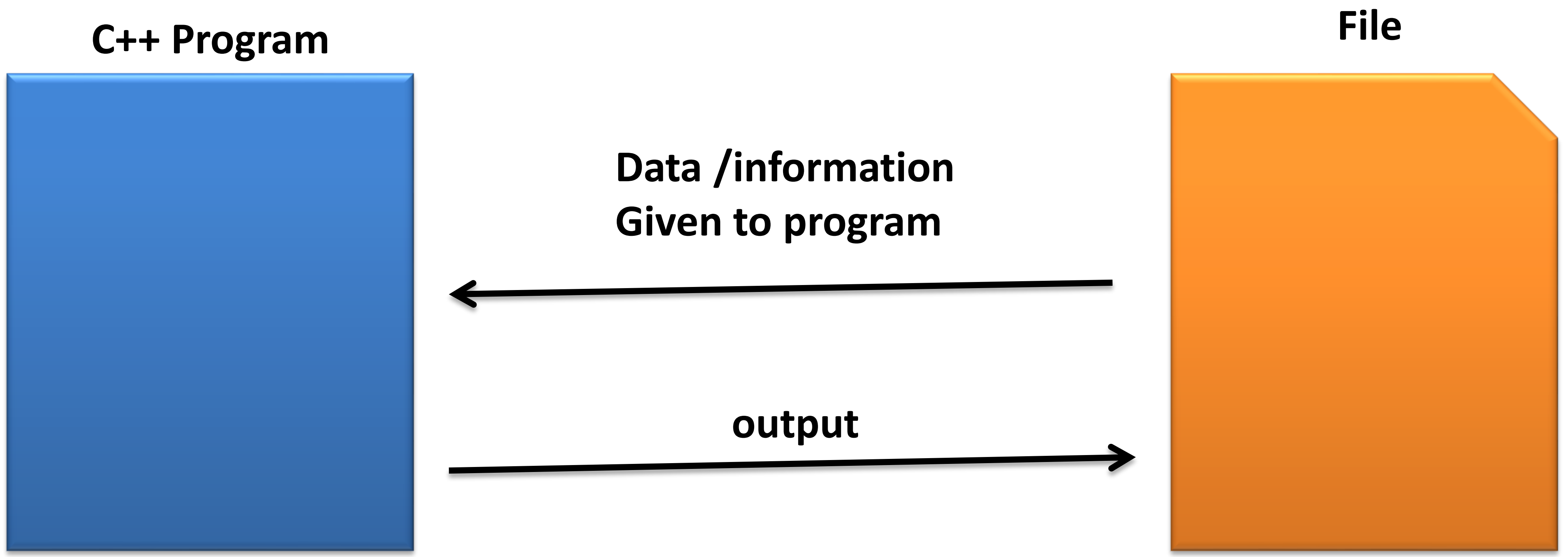
OPENTECHZ

A dream to realize your dream

File Handling

Opentechz Pvt Ltd .
By Parthasarathi Swain



Input : Data /information Given to program

Output : Data /information Given by program

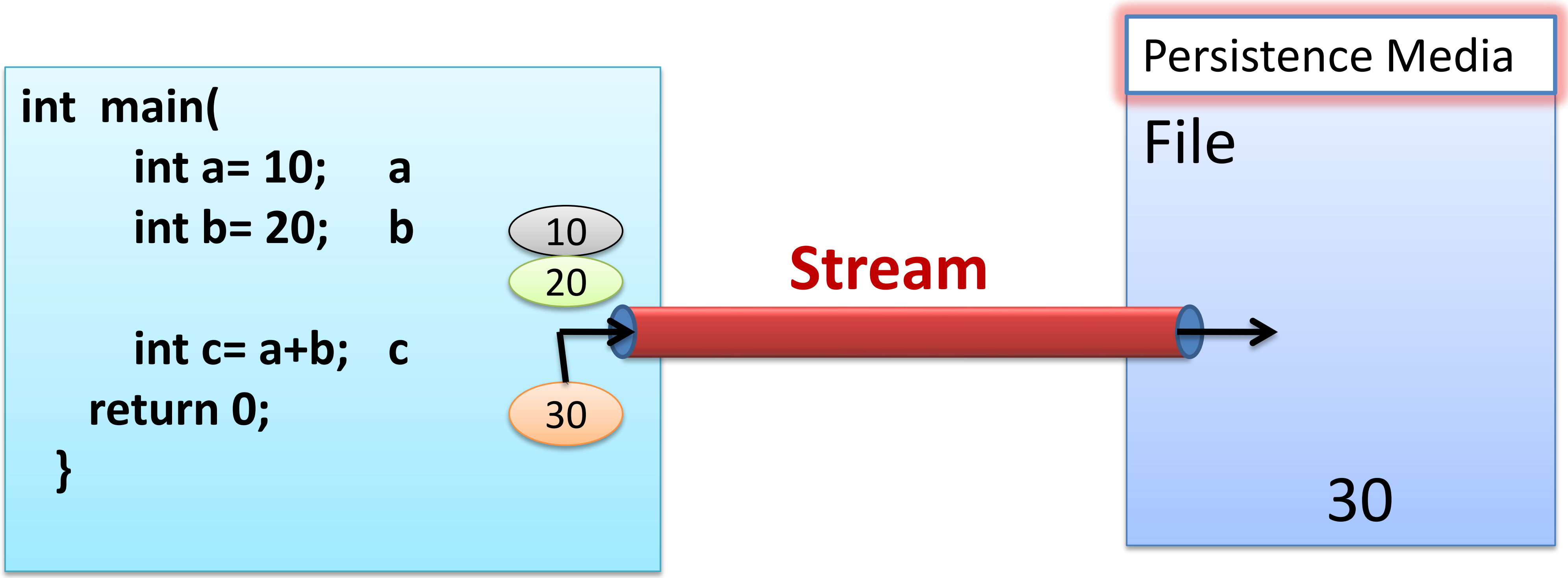
Introduction To Stream

Stream is a logical connection between c++ program and a file.

Stream can be defined as

" It is a continuous flow of data between c++ program and persistence media "

.



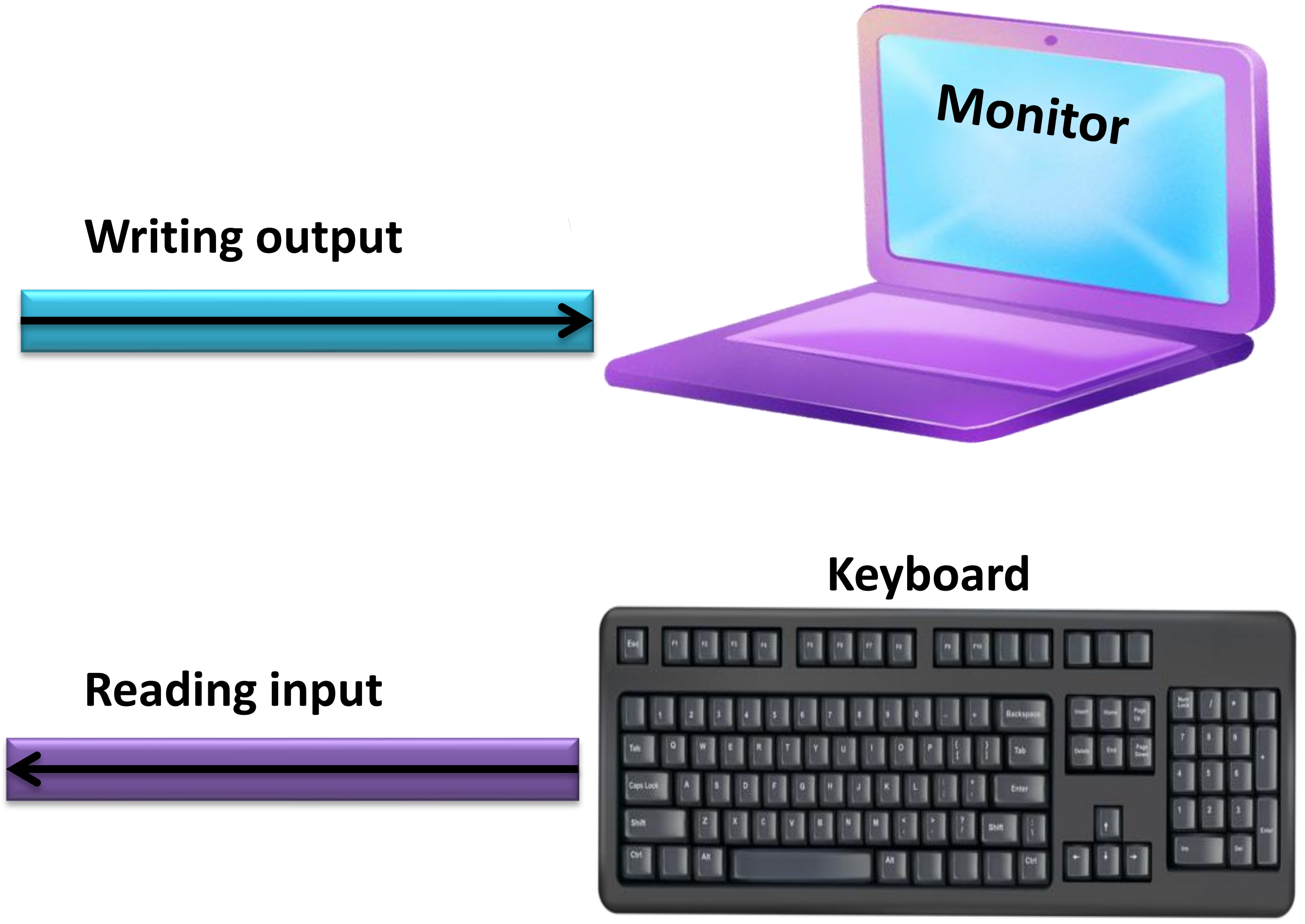
Stream is nothing but flow of data having Source and Destination .

```
int  main(
        int a= 10;

        int b= 20;

        int c= a+b;

    return 0;
  }
```



Stream is a class which provides operation & that Operation are called function .

For doing i/o operation c++ provides :

cout

cin

Cout (Console Output)

- It is a object type of Ostream class
- It perform formatted and unformatted output operation.
- It provides data/information to standard output device like : monitor
- It is available inside iostream
- This object uses << (insertion Operator) to perform o/p Operation.

Note :
<< it is a operator but it is Overloaded as a function of Ostream class. So we can access the member function through an object of Ostream class i.e cout .

Cin (Console Input)

- It is a object type of iostream class .
- It is used for reading data from standard input device like (keyboard).
- It performs formatted and unformatted input Operation.
- It does not require any format Specifiers .
- It uses >> (Extraction Operator) to extract data from keyboard.

Note :

>> Overloaded member function of istream class .

Introduction to File Handling in C++

Purpose: Allows programs to store and retrieve data **permanently** from storage.

File Types:

- **Text files** → Human-readable data (e.g., .txt)
- **Binary files** → Data in raw binary format (e.g., .dat)

Header File: <fstream> (File Stream)

Main Classes:

- **ofstream** → *Output File Stream* → Used to **write** to files.
- **ifstream** → *Input File Stream* → Used to **read** from files.
- **fstream** → *File Stream* → Used to **read and write** both.

Note :

 Tip: Think of a file like a notebook — **ofstream** writes in it, **ifstream** reads from it, and fstream does both.

Create and Write To a File

- To create a file, use either the ofstream or fstream class, and specify the name of the file.
- To write to the file, use the insertion operator (<<).

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

int main() {
  // Create and open a text file
  ofstream    MyFile(“otz.txt");

  // Write to the file
  MyFile << "Files can be tricky, but it is fun enough!";

  // Close the file
  MyFile.close();
}
```

Otz.txt

```
Files can be tricky, but it is
fun enough!
```

How it works:

ofstream MyFile("otz.txt"); → Creates/opens a file named otz.txt for writing.

MyFile << ...; → Writes text into the file.

MyFile.close(); → Closes the file to free resources.

⍰ If otz.txt does not exist, it will be created. If it exists, it will be overwritten.

Read a File

To read from a file, use either the ifstream or fstream class, and the name of the file.

Note that we also use a while loop together with the getline() function (which belongs to the ifstream class) to read the file line by line, and to print the content of the file:

Read a File

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

int main() {
    // Create a text string to store each line
    string myText;
    // Open the file for reading
    ifstream MyReadFile(“otz.txt");
    // Read the file line by line
    while (getline(MyReadFile, myText)) {
        // Output the text from the file
        cout << myText << endl;
    }
    // Close the file
    MyReadFile.close();
    return 0;
}
```

Otz.txt

Files can be tricky,
but it is fun enough!

How it works:

ifstream opens the file for reading.

getline() reads each line into myText.

while loop runs until the end of the file.

cout prints the contents line by line.

close() frees the file resource.

� This works well for text files. For binary files, we’d use read() instead of getline().