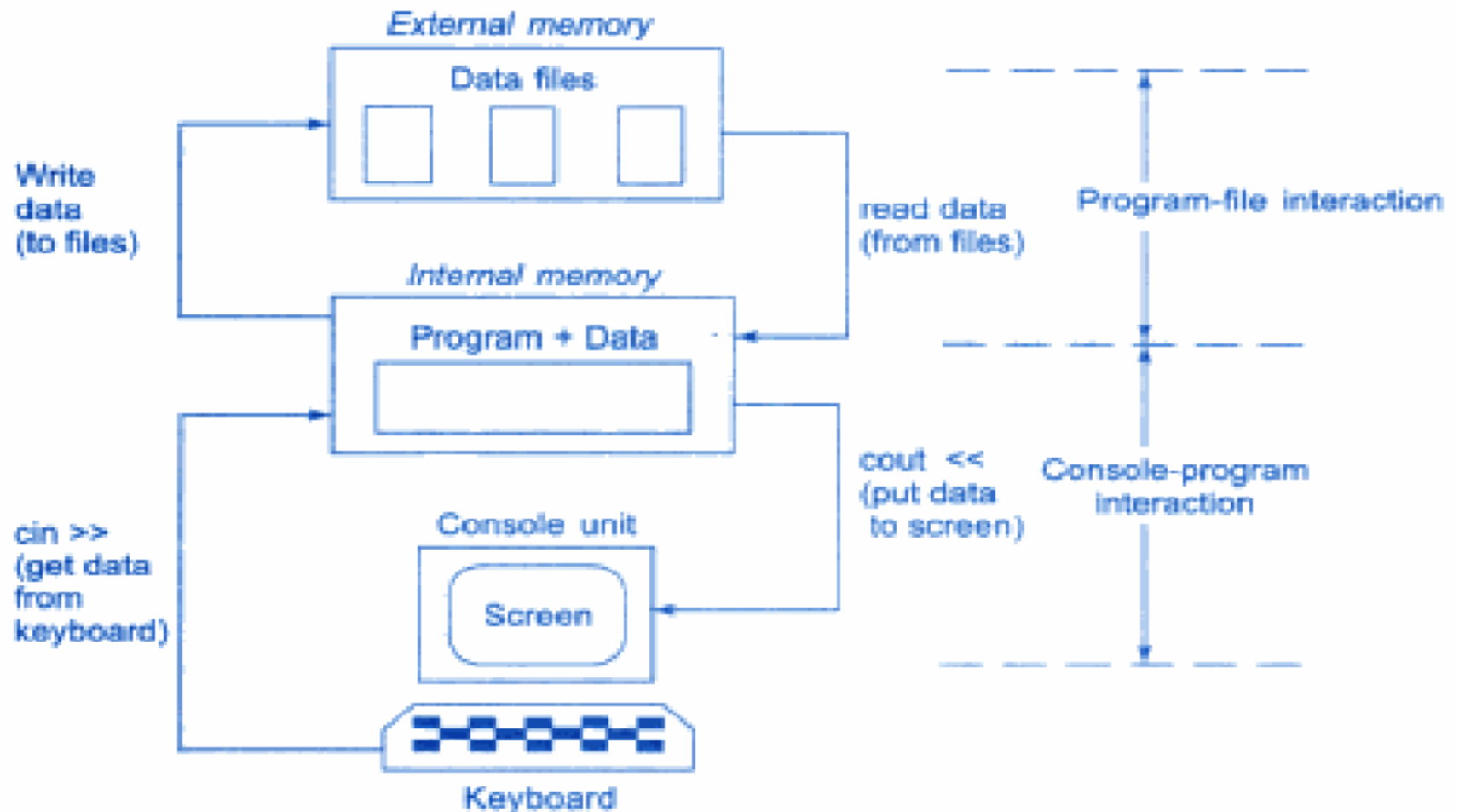


Working with Files

Introduction

- Many real-life problems handle large volumes of data.
- The data is stored in the devices using the concept of files.
- A file is a collection of related data stored in a particular area on the disk.
- Programs are designed to perform read and write operations on these files.

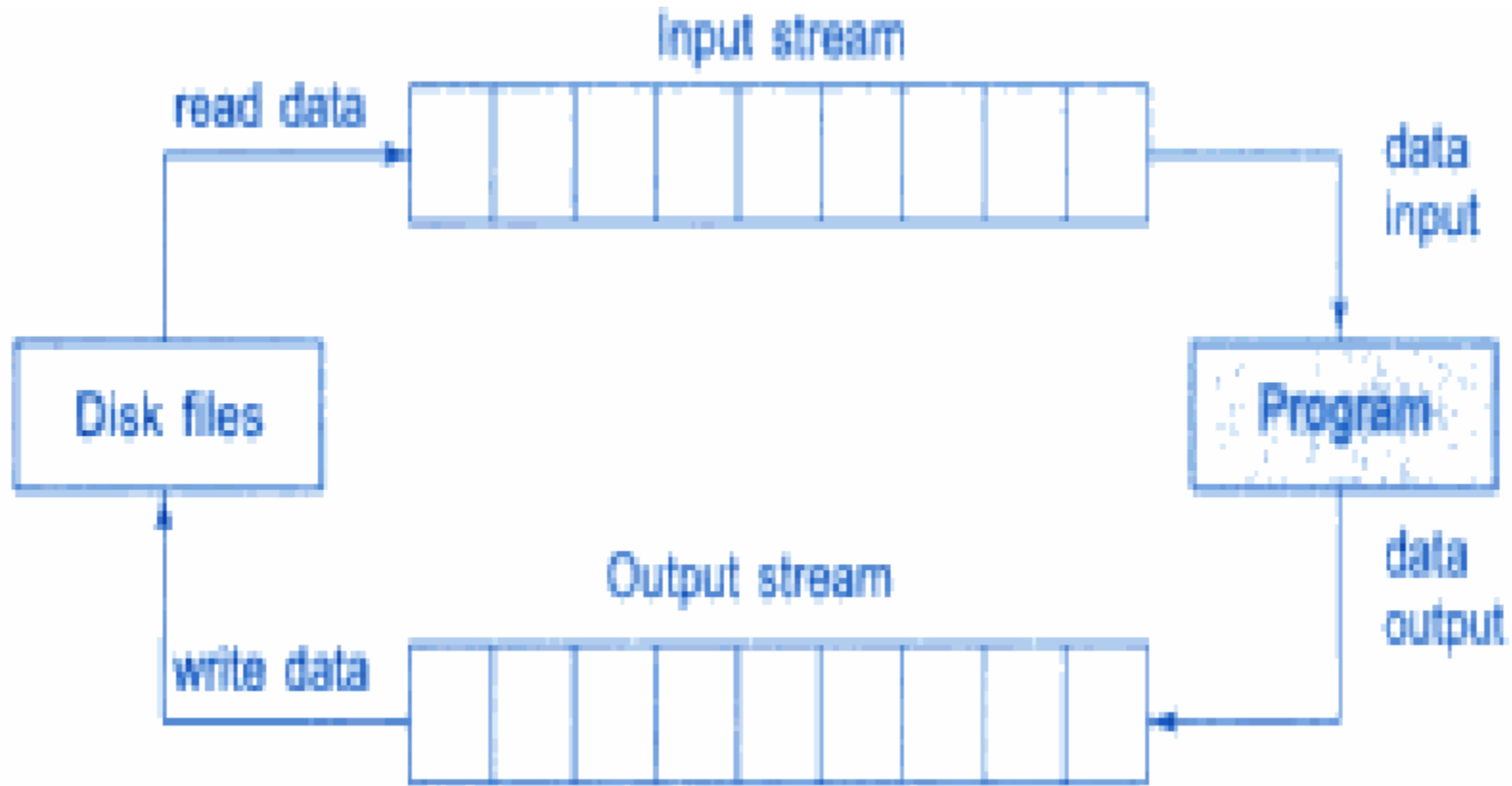
Console-Program-File interaction



Program File Communication

- In C++ **file streams** are used as an interface between the program and the files.
- The stream that supplies data to the program is known as **input stream** and the one that receives data from the program is known as **output stream**.
- Input stream => reads data
- Output stream => writes data

File input and Output Streams



File stream classes

filebuf	Its purpose is to set the file buffers to read and write. Contains Openprot constant used in the open() of file stream classes.Also contain close() and open() as members.
fstreambase	Provides operations common to file streams.Serves as a base for fstream,ifstream and ofstream class.Contains open() and close() functions.
ifstream	Provides input operations. Contains open() with default input mode.Inherits the functions get(),getline(),read(),seekg(),tellg() functions from istream.
ofstream	Provides output operations.Contains open() with default output mode.Inherits put(),seekp(),tellp() and write() functions from ostream.
fstream	Provides support for simultaneous input and output operations.Contains open with default input mode.Inherits all the functions from istream and ostream classes through iostream.

Opening and Closing a File

- To open a file, a file stream is created and then it is linked to the filename.
- A file can be opened in two ways:
 - Using the constructor function of the class.
 - Using the member function `open()` of the class.
- A file is closed by using the function `close()`.
- eg: `outfile.close();`

Opening File using Constructor

- Filename is used to initialize the file stream object.
- Create a file stream object using appropriate class.
 - `ofstream` class is used to create output stream.
 - `ifstream` class is used to create input stream.
- Initialize the file object with the desired filename.
- Eg:

`ofstream outfile ("results");` `//output only`

`ifstream infile ("data");` `// input only`

- `ofstream outfile (“results”);`
 - Creates *outfile* as an object of *ofstream* class that manages the output stream
 - This statement opens the file *results* and attaches it to the output stream *outfile*
- `ifstream infile (“data”);`
 - Creates *infile* as an object of *ifstream* class that manages the input stream
 - This statement opens the file *data* and attaches it to the input stream *infile*

- A program may contain statements like :
 - `outfile << "TOTAL";`
 - `outfile << sum;`
 - `infile >> number;`
 - `infile >> string;`

- We can use the same file for reading and writing data as follows:

Program1

.....

.....

ofstream outfile ("salary"); //creates outfile and connects salary to it

.....

.....

Program 2

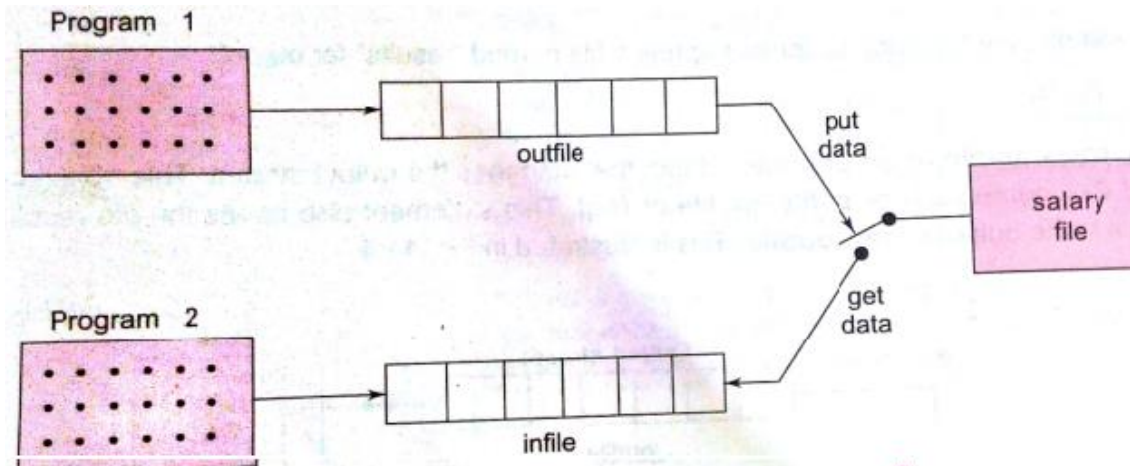
.....

.....

ifstream infile ("salary"); //creates infile and connects salary to it

.....

.....



- The connection with a file is closed automatically when the stream object expires i.e when a program terminates.
- In the above statement ,when the program 1 is terminated, the salary file is disconnected from the outfile stream.
- Instead of two programs, a single program can be used to do both reading and writing operations on a file as follows:

.....

.....

```
outfile.close(); //disconnect salary from outfile and connect to infile  
ifstream infile ("salary");
```

.....

.....

```
infile.close();
```

Opening File using Constructor

Program to use a single file for writing and reading the data

```
#include<iostream.h>
#include<fstream.h>
int main()
{
    ofstream  outf("Item");
    cout<<"Enter item name:";
    char name[30];
    cin >> name;

    outf<<name;
    cout<<"Enter item cost:";
    float cost;
    cin >> cost;

    outf << cost;
    outf.close();    //Disconnect the ITEM file from outf
```

Opening File using Constructor

```
ifstream inf("Item");    //Connect the ITEM file from inf

inf >> name;
inf >> cost;

cout << "Item name :" << name;
cout << "Item cost :" << cost ;

inf.close();
return 0;
}
```

Output:

Enter item name: CD-ROM

Enter item cost: 250

Item name: CD-ROM

Item cost: 250

Note that:

- When a file is opened for writing only, a new file is created if there is no file of that name.
- If a file by that name exists already , then its contents are deleted.

Opening Files Using open()

- The function open() can be used to **open multiple files** that use the same stream object.
- Syntax:

file-stream-class stream-object;

stream-object.open("filename");

- A stream object can be connected to only one file at a time.

Opening Files Using open()- Working with multiple files

```
#include<iostream.h>
#include<fstream.h>

int main()
{
    ofstream fout;                //create output stream
    fout.open("country");         //connect country to it

    fout<<"United state of America";
    fout<<"United Kingdom";

    fout.close();                 //disconnect country

    fout.open("capital");         //connect capital

    fout<<"Washington";
    fout<<"London";

    fout.close();                 //Disconnect capital
```

Opening Files Using open()- Working with multiple files

```
const int N=80;
char line[N];

ifstream fin;                //create input stream
fin.open("country");         //connect country

cout<<"Contents of country file" ;
while(fin)                   //check end of file
{
    fin.getline(line, N);
    cout<<line;
}
fin.close();                 //disconnect country
```

Opening Files Using open()- Working with multiple files

```
fin.open("Capital");           //connect capital

cout<<"Contents of capital file";

while(fin)                     //check eof
{
    fin.getline(line, N);
    cout<<line;
}
fin.close();                   //Disconnect capital
return 0;
}
```

- A stream can be connected to only one file at a time.
- Hence first file is closed before opening second one.

