Unit- V

Files & Streams

Unit - V

Notes By :- Mokashi Sir

Page No. : 01
Date : 30 / 10 / 18

Files & Streams

Introduction to Stream :-

Every Program takes some data as input and generates processed data as output by following the familiar input - process - output cycle. It is therefore, essential to know how to provide the input data and how to present the results in the desired form.

We have already used cin & cout with the >> and << operators for the input and output operations.

C++ supports a rich set of I/o functions and operations to do various manipulations. Since these functions use the advanced features of C++ (such as classes, derived classes and virtual functions) we need to know a lot about them before really implementing the C++ I/o operations.

Remember that C++ Support all of C's rich set of I/o functions. C++ also uses the concept of Stream & Stream Classes to implement its I/o operations with the Console and disk files. Stream classes Strongly Supports the Console oriented input-output Operation.

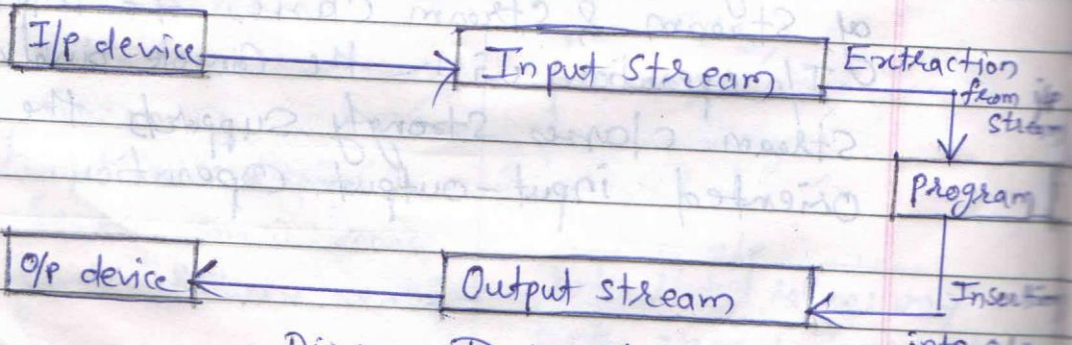
Notes By: Mokashi Sir

Page No.: 02
Date: / /

Concept of C++ Streams:-

The I/O system in C++ is designed to work with a wide variety of devices including terminals, disks and tape drives. Although each device is very different, the I/O system supplies an interface to the programmer that is independent of the actual device being accessed. This interface is known as stream.

Defn A stream is a sequence of bytes. It acts either as a 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 a program is called input stream. While the destination stream that recieves output from the program is called the output stream. In other words, a program extracts the bytes from an input stream and inserts bytes into an output stream.



Dia:- Data Streams

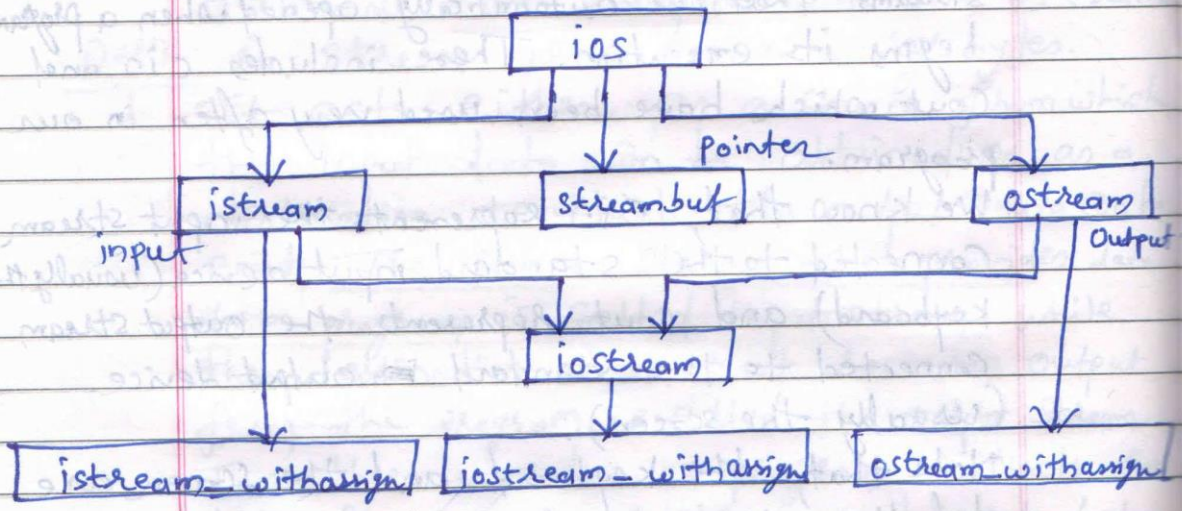
The data in the input stream can come from the keyboard or any other storage device. Similarly, the data in the output stream can go to the screen or any other storage device. Simply we can say that, a stream acts as an interface between the program and the input/output devices. C++ contains several pre-defined streams that are automatically opened when a program begins its execution. These includes cin and cout which have been used very often in our program.
We know that cin represents the input stream connected to the standard input device (usually the keyboard) and cout represents the output stream connected to the standard output device (usually the screen).
Note that, the keyboard and the screen are default options.

C++ Stream Classes:-

The C++ I/O system contains a hierarchy of classes that are used to define various streams to deal with both the console and disk files. These classes are called Stream classes.

3M* **Hierarchy of Stream Classes:-**

The hierarchy of the stream classes used for input and output operations with the console unit. These classes are declared in the header file iostream. This file should be included in all the programs that communicate with the console unit.



Dia:- Stream classes for Console I/O operations

From the above diagram, ios is the base class for istream (input stream) and ostream (output stream) which are, in turn, base classes for iostream (input/output stream). The class ios is declared as virtual base class so that only

One copy of its members are inherited by the iostream.

The class ios provides the basic support for formatted and unformatted I/O operations. The class istream provides the facilities for formatted and unformatted input while the class ostream (through inheritance) provides the facilities for formatted output.

~~The class iostream~~

Stream classes for Console Operation (Table found).

The class iostream provides the facilities for handling both input and output streams. Three classes, namely, istream_withassign, ostream_withassign & iostream_withassign add assignment operators to these classes.

Following table gives the details of these classes.

Name	Contents
	i) Contains basic facilities that are used for all other input and output classes.
class	ii) Also contains a pointer to a buffer object (streambuf object)
	iii) Declares constants and functions that are necessary for handling formatted input & output operations.

ClassName	Contents
istream (input stream)	i) Inherits the properties of ios. ii) Declares input functions such as get() & re iii) Contains overloaded extraction operator
Ostream (output stream)	i) Inherits the properties of ios. ii) Declares the output functions put() & writ iii) Contains overloaded insertion operator <<
iostream (input/output stream)	i) Inherits the properties of ios istream & ostream t multiple inheritance and thus contains all input and output functions.
streambuf	i) Provides an interface to physical devices thro buffers. ii) Acts as a base for filebuf class used

Opening and Closing a file :-

If we want to use a disk file, we to decide the following things about the fi and its intended use :

1. Suitable name for the file.
2. Data type and Structure.
3. Purpose.

4. Opening method.

The filename is a string of characters that make up a valid file name for the operating System. It may contains two parts, a primary name and an optional period with extension.

For opening a file, we must first create a file stream and then link it to the filename. A file stream can be defined using the classes ifstream, ofstream and fstream that are contained in the header file fstream. The class to be used depends upon the purpose, that is, whether we want to read data from the file or write data into it.

The connection with a file is closed automatically when the stream object expires (when the program terminates).

If we perform two different programs for writing data (output) and reading data (input), instead of these we can use a single program to do both the operations on a file.

For opening every file we can use the member function open() and for closing the file at the end of the program we can use the member function close() of fstream class.

3M * **File Modes:-**

The file mode parameter can tak one (or more) constants defined in ios cl Following table shows the file mode parameters with their meanings.

Parameter	Meaning
ios::app	Append to end-of-file
ios::ate	Go to end-of-file on open
ios::binary	Binary file
ios::in	Open file for reading only
ios::nocreate	Open files if the file does no exist
ios::noreplace	Open file if the file alread exist
ios::out	Open file for writing only
ios::trunc	Delete the contents of the file if it exists

File I/O with stream class:-

The I/O system of C++ contains a set of classes that defines the file handling metho There include ifstream, ofstream and fstream These classes are derived from the corresponi iostream class.

These classes designed to manage the disk files, are declared in fstream and therefore, we must include this file in any program that uses files.

Following table shows the details of file stream classes. These classes contains many more features.

Class	Contents
filebuf	Its purpose is to set the file buffers to read and write. It also contains close() and open() as members.
streambase	It provides operations common to the file streams. It serves as a base for fstream, ifstream and ofstream class.
ifstream	It provides input operations. It contains open() with default input mode.
ofstream	It provides output operations. It contains open() with default output mode.
fstream	It provides support for simultaneous input and output operations. It contains open() with default input mode.

Best of Luck.

Question Bank on Unit-V

Q1. What is Polymorphism? Explain its different types?

Q2. What is Virtual Function? Explain rules for Virtual Function.

Q3. Explain Virtual Function with suitable program.

Q.4 Explain Pure Virtual function with suitable program.

Q.5 Write difference between Polymorphism and Virtual function.

Q.6 Write difference between Virtual Function and Pure Virtual Function.

Q.7 What is mean by streams? Explain hierarchy of stream classes.

Q.8 Explain various stream classes used for console operation.

Q.9 Explain different file modes.

Q.10 Explain file i/o with Stream class.

****