Dr. SNS RAJALAKSHMI COLLEGE OF ARTS AND SCIENCE
(Autonomous)
Coimbatore-641049.
Accredited by NAAC (Cycle-III) with 'A+' Grade
(Recognised by UGC, Approved by AICTE, New Delhi and
Affiliated to Bharathiar University,Coimbatore)

C++ FILES AND STREAMS

Course Name: Object Oriented Programming
Course code: 21UCU403
UNIT: II
Prepared By : Dr.A.DEVI

C++ FILES AND STREAMS

The **iostream** standard library, which provides **cin** and **cout**methods for reading from standard input and writing to standard output respectively.

How to read and write from a file.

This requires another standard C++ library called **fstream**, which defines three new data types:

DataType	Description
ofstream	This datatyperepresents theoutput filestreamandis usedto create files and to write information to files.
ifstream	This datatyperepresents theinput filestreamandis usedtoread information from files.
fstream	This data type represents the file stream generally, and has the capabilities of both ofstream and ifstream which means it can createfiles,writeinformationto files,andreadinformationfrom files.

To perform file processing in C++,header files<iostream> and <fstream> mustbe included in your C++ source file.

OpeningaFile:

A file mustbe opened before you can read from itor write to it.Either the **ofstream** or **fstream** objectmaybe used to open a file for writing and ifstream objectisused to open a file for reading purpose only.

Following isthe standard syntaxfor open function,which isa member of fstream,ifstream,and ofstream objects.

```
voidopen(constchar*filename,ios::openmodemode);
```

Here,the firstargumentspecifiesthe name and location of the file to be opened and the second argument of the**open** member function defines themodein which thefileshould beopened.

ModeFlag	Description
ios::app	Appendmode.Alloutputtothatfiletobeappendedtotheend.
ios::ate	Open afileforoutput and movetheread/writecontrol totheend of the file.
ios::in	Openafileforreading.
ios::out	Openafileforwriting.
ios::trunc	If the file already exists, its contents will be truncated before opening the file.

Youcancombinetwoormoreofthesevaluesby**OR**ingthemtogether.Forexampleifyouwantto openafileinwritemodeandwant totruncateit incaseit alreadyexists, followingwillbethe syntax:

```
ofstreamoutfile;
outfile.open("file.dat",ios::out|ios::trunc);
```

Similarway,youcanopenafileforreadingandwritingpurposeas follows:

```
fstreamafile;
afile.open("file.dat",ios::out|ios::in);
```

ClosingaFile

When a C++ programterminates it automatically closes flushes all thestreams, releaseall the allocated memory and close all the opened files. But it is always a good practice that a programmer should closeall theopened files beforeprogramtermination.

Following isthe standard syntaxfor close function,which isa member of fstream,ifstream,and ofstream objects.

```
voidclose();
```

WritingtoaFile:

While doing C++ programming, you write information toafile from your program using the stream insertion operator<<justasyou use thatoperator to outputinformation to the screen. Theonly differenceis that you usean **ofstream**or**fstream**object instead of the**cout**object.

ReadingfromaFile:

You read information from a file into your program using the stream extraction operator>>just as you use that operator to input information from the keyboard. The only difference is that you use an **ifstream** or **fstream** object instead of the **cin** object.

Read&WriteExample:

Following is the C++ program which opens a file in reading and writing mode. After writing information inputted bythe user to a file named afile.dat,the program readsinformation from the file and outputs it onto the screen:

```
#include <fstream>
#include <iostream>
usingnamespacestd;

int main()
{

   Char data[100];

   // open a file in write mode.
   ofstream outfile;
   outfile.open("afile.dat");

   cout<<"Writing to the file"<<endl; cout
   <<"Enter your name: "; cin.getline(data,
   100);

   // write inputted datainto the file.
   outfile <<data <<endl;

   cout<<"Enter your age:"; cin
   >>data;
   cin.ignore();

   // againwrite inputted datainto the file. outfile
   <<data <<endl;

   // close the opened file.
   outfile.close();
```

```
      ifstream infile;
      infile.open("afile.dat");

      cout <<"Reading from the file"<<endl;
      infile >>data;

      // write the dataatthe screen. cout
      <<data <<endl;

      // againread the datafrom the file and display it. infile
      >>data;
      cout<<data<<endl;

      // close the opened file.
      infile.close();

      return0;
   }
```

When the above code is compiled and executed, it produces the following sample input and output:

```
$./a.out
Writing to the file
Enter your name: Zara
Enter your age: 9
Reading from the file
Zara
9
```

Above examplesmake use of additional functionsfrom cin object,like getline function to read the line from outside and ignore function to ignore the extra characters left by previous read statement.

FilePositionPointers:

Both **istream** and **ostream** provide member functions for repositioning the file-position pointer.These member functionsare **seekg**"*seekget* "for istream and **seekp**"*seekput* "for ostream.

Theargument toseekgandseekpnormally is alonginteger. Asecondargument canbespecified to indicate the seek direction.The seek direction can be **ios::beg** *thedefault*for positioning relative to the beginning of a stream,**ios::cur** for positioning relative to the currentposition in a stream or **ios::end** for positioning relative to the end of a stream.

The file-position pointer is an integer value thatspecifies the location in the file as a number of bytes from the file's starting location.Some examples of positioning the "get" file-position pointer are:

```
// positionto the nthbyte of fileObject(assumesios::beg)
fileObject.seekg( n );
```

```
// position n bytes forward in fileObject
fileObject.seekg( n, ios::cur );
```

```
// position n bytes back from end of fileObject
fileObject.seekg( n, ios::end );
```

```
// positionatend of fileObject
fileObject.seekg( 0, ios::end
);
```