21

FILES

- **FILES:** A file is a collection of logically related records. A program usually requires two types of data communication:-
  - writing data on data file
  - reading data from data file

- **HOW TO USE FILES? :**

In C++, files are mainly dealt by using three classes fstream, ifstream, ofstream available in fstream header file.

ofstream: Stream class to write on files
ifstream: Stream class to read from files
fstream: Stream class to both read and write from/to files.

- **WRITING DATA ON THE FILE:** The data flows from keyboard to memory and from memory to storage device.

Keyboard → memory → hard disk/ storage device

- **READING DATA FROM THE FILE:** The data flows from storage device to memory and from memory to output device, particularly monitor.

  Data file → memory → output device (screen) or external storage device (hard disk/storage device)

- **OPENING A FILE:** A file can be opened in two ways:
  - Using constructor function of a class.
  - Using the member function open ( ) of the class

- The following statement opens the file STU.DAT in output mode, i.e., for writing data on the file.

ofstream outfile ("STU.DAT");

Where **ofstream** is a class and **outfile** is user defined object.

- **outfile** < < "TOTAL MARKS" << "\n";
  outfile < < total << "\n";

The statements are used for writing data on the file. The newline character is used for moving the pointer to the next line.

- **ifstream** infile ("STU.DAT");

  opens the file "STU.DAT" in input mode, i.e., for reading purpose .

- The statements infile >> string; infile >> number; read the data from the data file.

PROGRAM:

/* File Handling with C++ using ifstream & ofstream class object*/

```
#include <iostream>

/* fstream header file for ifstream, ofstream,
  fstream classes */
#include <fstream>

using namespace std;

// Driver Code
int main()
{
    // Creation of ofstream class object
    ofstream fout;
    string line;
    fout.open("sample.txt");

    // Execute a loop If file successfully
opened
    while (fout) {

        // Read a Line from standard input
        getline(cin, line);

        // Press -1 to exit
        if (line == "-1")
            break;

        // Write line in file
        fout << line << endl;
    }

    // Close the File
    fout.close();

    // Creation of ifstream class object to read
the file
    ifstream fin;

    fin.open("sample.txt");

    // Execute a loop until EOF (End of File)
    while (fin) {

        // Read a Line from File
        getline(fin, line);

        // Print line in Console
        cout << line << endl;
    }

    // Close the file
    fin.close();

    return 0;
}
```

- **OPENING A FILE USING OPEN()** :
  - First a stream object is assigned to and then it is used to open the file in turn.

SYNTAX:

filestream_class stream_object;
stream_object . open (“filename”);

For example :

ofstream outfile;
outfile . open (“ABC”);
outfile . close ( );
outfile . open (“XYZ”);
outfile.close ( )

- The open () function has two parameters :
  SYNTAX:
  stream_object . open (“filename”, access mode);
  - ✓ ios : : in for ifstream functions
  - ✓ ios : : out for ofstream functions

- **WRITE() & READ() FUNCTION :**

The functions write ( ) and read ( ) have two parameters: address of the variable, size of the variable.

SYNTAX:

infile . read ((char*) & v, sizeof v);
outfile . write ((char*) & v, sizeof v); where v is the variable.

- **FILE POINTERS**:

File has two associated pointers called input pointer (or get pointer) and output pointer (or put pointer).

  - **seekg** ( ) It moves get pointer to a specified location.
  - **seekp** ( ) It moves the put pointer to a specified location.
  - **ios:: beg – Beginning of the file**
  - **ios:: cur – Current position of the pointer**
  - **ios:: end – End of the file**
  -

- **tellg() & tellp() FUNCTION:**
  tellg ( ) - Gives the position of get pointer in terms of number of bytes.
  tellp ( ) - Gives the position of put pointer in terms of bytes.

- **CLOSE () FUNCTION**:

stream_object.close ( )

- **PROGRAM:**
  The following program to
  Create a data file
  Display a data file
  Adding a new record
  Modify the existing record

```
# include
class student
{
char name [30];
int rn;
public: void getdata ( );
void putdata ( );
};
void student : : getdata ( )
 { cout <> name; cout << “Enter roll number”; cin >> rn;
}

void student :: putdata ( )
{
cout << “Student name” << name << “\n”;
cout << “Student roll number” << rn << “\n”;
}
void main ( )
{
fstream file;
file . open ( “ABC”, ios::in | ios::out | ios::binary);
student st;
int i, n;
cout << “How many records to enter”;
cin >> n;
for (i = 1; i < = n, i ++ )
{
st. getdata ( );
file . write ((char*) & st, sizeof st);
} / / Display a data file
file . seekg ( 0, ios::beg);
while (file . read ((char*) & st, sizeof st))
{
st. putdata ( );
}
file . clear ( ); / / To make the end of file mark false
// To append record
st . getdata ( );
file . write ((char*) & st, sizeof st);
// To modify a record
file.clear ( );
cout << “Enter record number”;
cin >> n;
file . seekp ((n - 1)* sizeof st, ios::beg);
st. getdata ( );
 file.write ((char*) & st, sizeof st);
// To close a file
file . close ( );
}
```

CHECK YOURSELF

1. Which header file is required to use file I/O operations?
A) <ifstream>
B) <ostream>
C) <fstream>
D) <iostream>

2. Which of the following is not used as a file opening mode?
A) ios::trunc
B) ios::binary
C) ios::in
D) ios::ate

3. By default, all the files in C++ are opened in _________ mode.
A) Text
B) Binary
C) ISCII
D) VTC

4. What is the return type open() method?
   A) int
   B) char
   C) bool
   D) floatis a pointer to a string
5. Which operator is used to insert the data into file?

   A) >>
   B)<<
   C) <
   D) None of the above

ANSWERS

Answers to Check Yourself:

1. C
2. B
3. A
4. C
5. B

STRETCH YOURSELF

1. Write a program to write and read from file?
2. Write a program using open() to read and write a file.
3. Write a program to enter student details in the file and display the output.