



OOP

**Modern College of Engineering**

Shivajinagar, Pune 5. Roll no: 21027

Name- Mahesh Jagtap

## Assignment no.4

Title : File handling

Problem Statement :

write a C++ program that creates an output file, writes information to it, closes the file & open it again as an input file & read the information from the file.

Prerequisites : object oriented programming

Objective :-

To learn the Concept of file handling.

Theory :

Files represents storage medium for storing data & information.

Streams refers to sequence of bytes. In files we store data i.e. text or binary data permanently & use these data to read or write in the form of input output operations by transferring bytes of data. So we use the term file streams/ file handling. We use the header file `<fstream>`



## 1. InputStream

These are used to hold input from a data producer, such as a keyboard, a file, or a network.

The source stream that provides data to the program is called the input stream. A program that extracts the bytes from the input stream

In most cases the standard input device is the keyboard. with the cin & "extraction" operator ( $\gg$ ) it is possible to read input from the keyboard.

## 2. OutputStream:

These are used to hold output for a particular data consumer, such as a monitor, a file or a printer. The destination stream that receives data from the program is called the output stream.

A program inserts the bytes into an output stream. By default, the std output of a program points at the screen. so, with the cout operator & the "insertion" operator ( $\ll$ ) \* you can print a message on the screen.

3



## Modern College of Engineering

Shivajinagar, Pune 5.

iostream standard library provides cin & cout methods for reading from standard input & writing to standard output respectively.

file handling provides three new datatypes:

1	sr. No.	Data type	Description
	1.	ofstream	This data type represents the output file stream & is used to create files & to write information to files.
	2.	ifstream	This data type represents the input file stream & is used to read information from files.
	3.	fstream	This data type represents the file stream generally, & has the capabilities of both ofstream & ifstream which means it can create files, write information to files, & read info from files.





To perform file processing in C++, header files `<iostream>` & `<fstream>` must be included in your C++ source file.

## ❖ Opening a file:

A file must be opened before you can read from it or write to it. Either `ofstream` or `fstream` object may be used to open a file writing. And `ifstream` object is used to open a file for reading purpose only.

Following is the std syntax for `open()` function, which is a member of `fstream`, `ifstream` & `ofstream` objects.

```
void open (const char *filename, ios::open-  
mode mode);
```

Here, the first argument specifies the name & location of the file to be opened & the second argument of the `open()` member function defines the mode in which the file should be opened.





## Modern College of Engineering

Shivajinagar, Pune 5.

Sr. No.	Mode flag	Description
1.	ios :: app	Append mode. All output to that file to be appended to the end.
2.	ios :: ate	open a file for output & move the read/write control to the end of the file.
3.	ios :: in	open a file for reading.
4.	ios :: out	open a file for writing
5.	ios :: trunc	if the file already exists, its contents will be truncated before opening the file.

You can combine two or more of these values by ORing them together. for example if you want to open a file in write mode & want to truncate it in case that already exists, following will be the syntax:

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





## Modern College of Engineering

Shivajinagar, Pune 5.

Similar way, you can open a file for reading & writing purpose as follows:-

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

### ❶ closing a file.

When a C++ program terminates it automatically flushes all the streams, release all the allocated memory & close the all opened files.

But it is always a good practice that a programmer should close all the opened files before program termination.

following is the std syntax for close() function, which is a member of fstream, ifstream, & ofstream objects.

```
void close();
```

❶ writing into file & closing the file

```
new_file << "learning file handling" ;  
           // writing to file  
new_file.close();
```



7



## Modern College of Engineering

Shivajinagar, Pune 5.

Here, we first create a new file "new\_file\_write" using `open()` function since we wanted to send output to the file so, we use `ios::out`. As given in the program, information typed inside the quotes after insertion pointer "<<" got passed to the output file. once the work is complete, we close the file.

### ② Reading from a file

- 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.

`file.read (char* &s, sizeof(s));`

- These functions take two arguments. The first is the address of the variable `v`, & the second is the length of that variable in bytes. The address of variable must be cast to type `char*` (ie. pointer to character type)



## ⑧ writing to a file

- while doing c++ programming, you write information to a file from your program using the stream insertion operator (`<<`) just as you use that operator to output information to the screen.
- The only diff. is that you can use `ofstream` or `fstream` object instead of `cout` object.

`file.write(Cchar*) &s, sizeof(s);`

## Algorithm:

1. start
2. create class
3. define data members roll no & name.
4. Define `accept()` to ~~make~~ take name & roll no. from user.
5. Define `display()` to display the record
6. In `main()`,  
    create the object of class & `fstream` class.
7. Take a limit from user in 'n' variable.
8. open the file in out mode, call `accept()` to take record from user, then call `write()` to write that record into the file & at the end close that file.





# Modern College of Engineering

Shivajinagar, Pune 5.

9

9. open the file in 'in' mode, read the record from the file, call display() function to display the record & at the end close that file.

10. stop.

I/p:

how many records you want : 3

1. abc

2. pqr

3. xyz

O/p:

name : abc

Roll No. 1

name : pqr

Roll No. 2

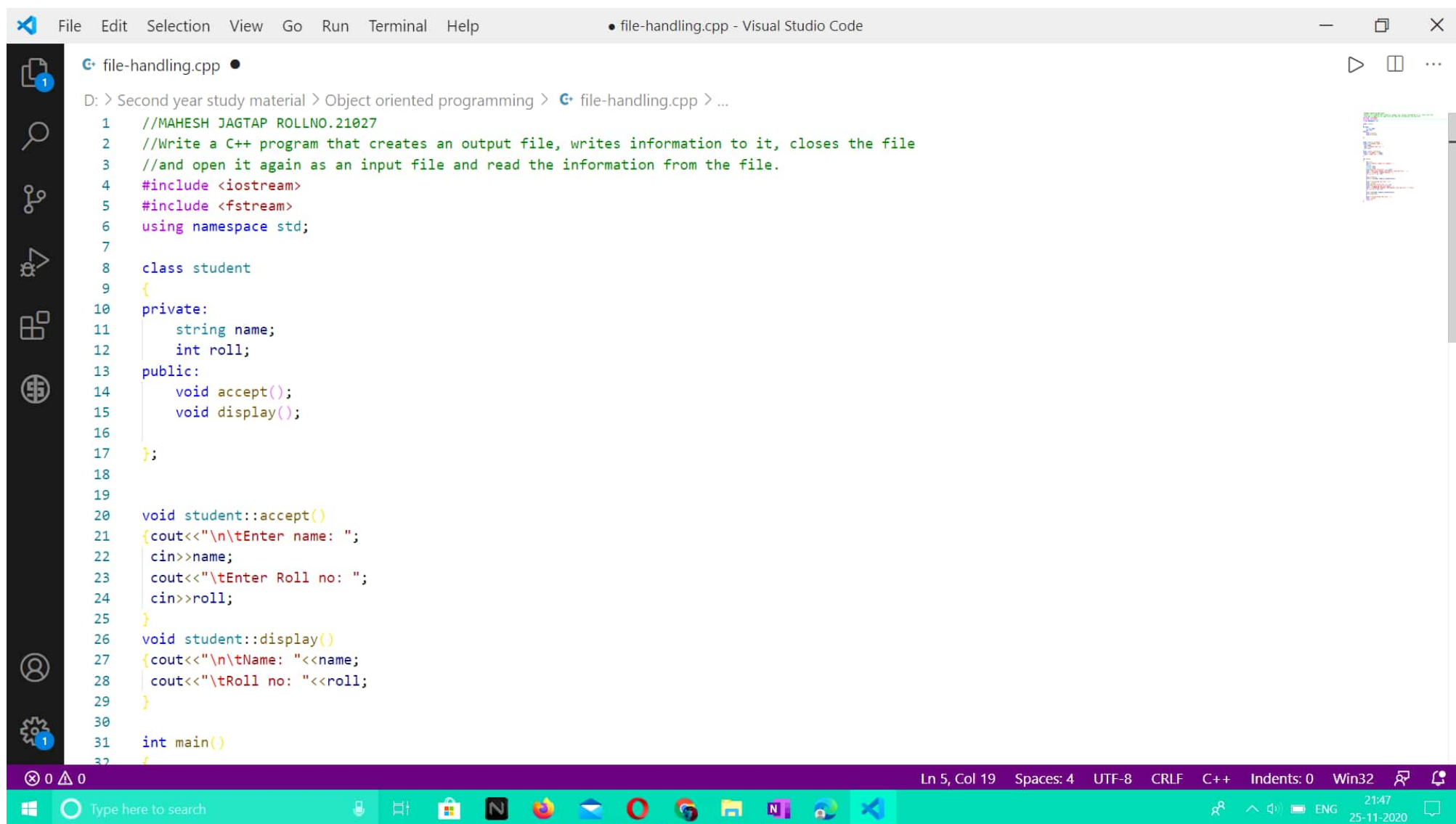
name : xyz

Roll No. 3

Conclusion:

Do, in this assignment, we studied diff. file handling operations and like, opening, closing, reading & writing and performed successfully on the files.





file-handling.cpp •

D: > Second year study material > Object oriented programming > file-handling.cpp > ...

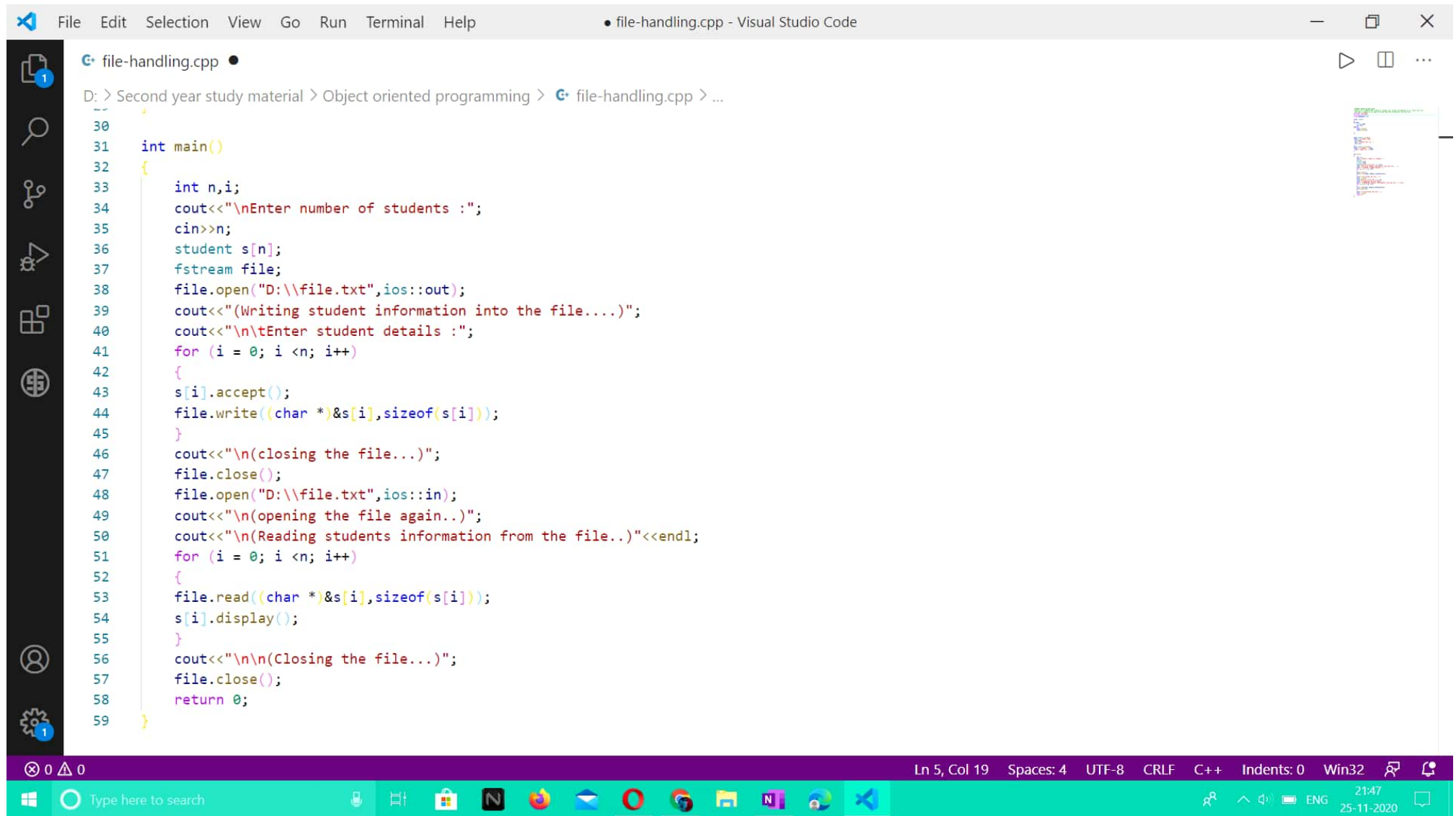
```
1 //MAHESH JAGTAP ROLLNO.21027
2 //Write a C++ program that creates an output file, writes information to it, closes the file
3 //and open it again as an input file and read the information from the file.
4 #include <iostream>
5 #include <fstream>
6 using namespace std;
7
8 class student
9 {
10 private:
11     string name;
12     int roll;
13 public:
14     void accept();
15     void display();
16
17 };
18
19
20 void student::accept()
21 {cout<<"\n\tEnter name: ";
22   cin>>name;
23   cout<<"\tEnter Roll no: ";
24   cin>>roll;
25 }
26 void student::display()
27 {cout<<"\n\tName: "<<name;
28   cout<<"\tRoll no: "<<roll;
29 }
30
31 int main()
32 {
```

Ln 5, Col 19 Spaces: 4 UTF-8 CRLF C++ Indents: 0 Win32

Type here to search

21:47 25-11-2020





```
file-handling.cpp
D: > Second year study material > Object oriented programming > file-handling.cpp > ...
30
31 int main()
32 {
33     int n,i;
34     cout<<"\nEnter number of students :";
35     cin>>n;
36     student s[n];
37     fstream file;
38     file.open("D:\\file.txt",ios::out);
39     cout<<"(Writing student information into the file....)";
40     cout<<"\nEnter student details :";
41     for (i = 0; i <n; i++)
42     {
43         s[i].accept();
44         file.write((char *)&s[i],sizeof(s[i]));
45     }
46     cout<<"\n(closing the file...)";
47     file.close();
48     file.open("D:\\file.txt",ios::in);
49     cout<<"\n(opening the file again..)";
50     cout<<"\n(Reading students information from the file..)"<<endl;
51     for (i = 0; i <n; i++)
52     {
53         file.read((char *)&s[i],sizeof(s[i]));
54         s[i].display();
55     }
56     cout<<"\n\n(Closing the file...)";
57     file.close();
58     return 0;
59 }
```

Ln 5, Col 19 Spaces: 4 UTF-8 CRLF C++ Indents: 0 Win32

Type here to search

21:47 25-11-2020





The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the execution of a C++ program named 'file-handling.cpp'. The program prompts the user to enter the number of students (2), then collects details for each student (name and roll number). It then closes the file, opens it again, and reads the stored information, displaying it on the screen. The terminal output is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Jagta> cd "d:\Second year study material\Object oriented programming\" ; if ($?) { g++ file-handling.cpp -o file-handling } ; if ($?) { .\file-handling }

Enter number of students :2
(Writing student information into the file....)
  Enter student details :
    Enter name: mahesh
    Enter Roll no: 21027

    Enter name: shreyas
    Enter Roll no: 21036

(closing the file...)
(opening the file again..)
(Reading students information from the file..)

    Name: mahesh      Roll no: 21027
    Name: shreyas     Roll no: 21036

(Closing the file...)
PS D:\Second year study material\Object oriented programming> cd "d:\Second year study material\Object oriented programming\" ; if ($?) { g++ file-handling.cpp -o file-handling } ; if ($?) { .\file-handling }
```