NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

CL 217 – Object Oriented Programing Lab

Lab 11

Outline

- Filing
- Examples
- Exercise

Introduction to filing

Files are used to store data in a storage device permanently.
File handling provides a mechanism to store the output of a program in a file and to perform various operations on it.
A stream is an abstraction that represents a device on which operations of input and output are performed.

Streams Operation

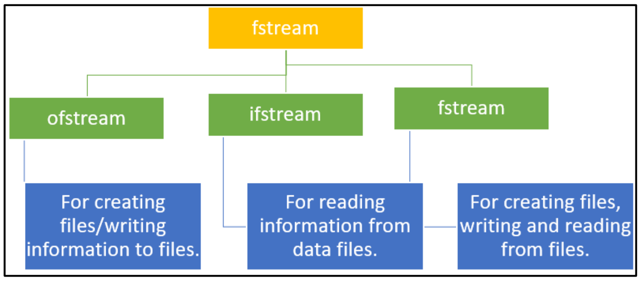
- ***Stream***
  - A transfer of information in the form of a sequence of bytes
- ***I/O Operations***
  - Input: A stream that flows from an input device (i.e.: keyboard, disk drive, network connection) to main memory
  - Output: A stream that flows from main memory to an output device (i.e.: screen, printer, disk drive, network connection)

Iostream Library Header Files

<iostream.h>: Contains cin & cout objects

File handling Classes

- ***These includes:***
  - ***ofstream***: This Stream class signifies the output file stream and is applied to create files for writing information to files
  - ***ifstream:*** This Stream class signifies the input file stream and is applied for reading information from files
  - ***fstream:*** This Stream class can be used for both read and write from/to files

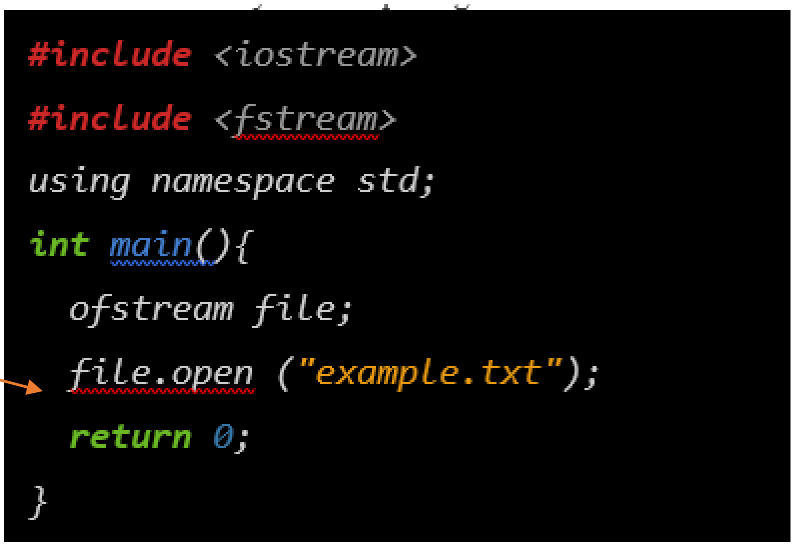

fstream
ofstream
ifstream
fstream
For creating files/writing information to files.
For reading information from data files.
For creating files, writing and reading from files.

Opening a File

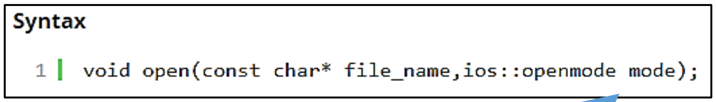
- We can open a file using any one of the following methods:
  - First is bypassing the file name in constructor at the time of object creation.
  - Second is using the open() function**.**

Ofstream are used to create and write the file

Open() function open your file



```
#include <iostream>
#include <fstream>
using namespace std;
int main(){
  ofstream file;
  file.open ("example.txt");
  return 0;
}
```



Syntax

```
void open(const char* file_name,ios::openmode mode);
```

The second argument represents the mode in which the file has to be opened

Opening Files Mode

1 **ios::app**
Append mode. All output to that file to be appended to the end.

2 **ios::ate**
Open a file for output and 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.

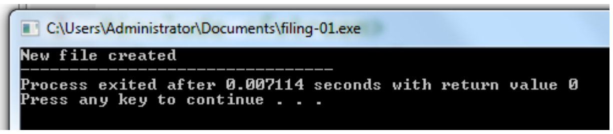
5 **ios::trunc**
If the file already exists, its contents will be truncated before opening the file.

File Processing Function

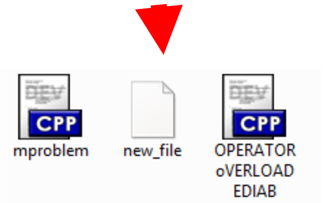
- ✓ open(): To create a file
- ✓ close(): To close an existing file
- ✓ get(): to read a single character from the file
- ✓ put(): to write a single character in the file
- ✓ read(): to read data from a file..>>
- ✓ write(): to write data into a file.. <<

Opening File- Example

```
#include<iostream>
#include <fstream>
using namespace std;
int main()
{
fstream FileObject;
FileObject.open("new_file",ios::out);
if(! FileObject.is_open())
{
cout<<"File creation failed";
}
else
{
cout<<"New file created";
new_file.close(); // Step 4: Closing file
}
return 0;}
```



```
New file created
--------------------------------
Process exited after 0.007114 seconds with return value 0
Press any key to continue . . .
```



Open()

- Opening a file associates a file stream variable declared in the program with a physical file at the source, such as a disk.
- In the case of an input file:
  - the file must exist before the open statement executes.
  - If the file does not exist, the open statement fails and the input stream enters the fail state
- An output file does not have to exist before it is opened;
  - if the output file does not exist, the computer prepares an empty file for output.
  - If the designated output file already exists, by default, the old contents are erased when the file is opened.

Validate the file before trying to access

Method 1:

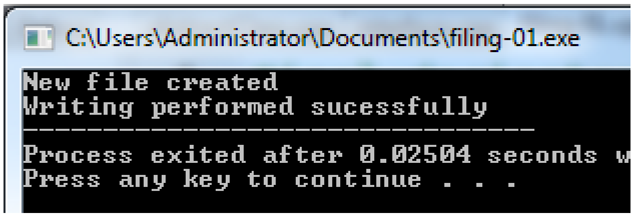
```
By checking the stream variable;
If ( ! Mystream)
{
Cout << “Cannot open file.\n ”;
}
```

Method 2:

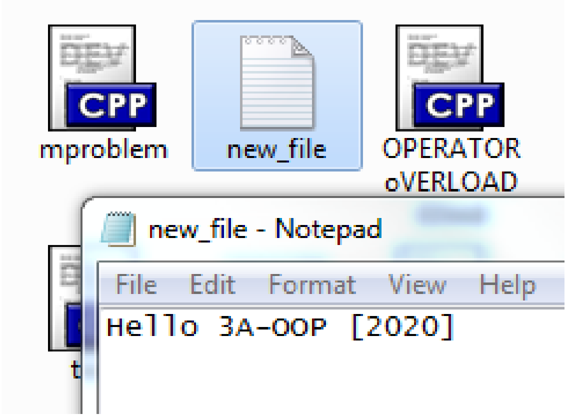
```
By using bool is_open() function.
If ( ! Mystream.is_open()) {
Cout << “File is not open.\n ”;
}
```

Writing File- Example

```
#include<iostream>
#include <fstream>
using namespace std;
int main(){
fstream new_file;
new_file.open("new_file",ios::out);
if(!new_file)
{
cout<<"File creation failed";
}
else
{
cout<<"New file created";
new_file<<" Hello 3A-OOP [2020]";
cout<<"Writing performed sucessfully";
new_file.close(); // Step 4: Closing file
}return 0;}
```



```
New file created
Writing performed sucessfully
--------------------------------
Process exited after 0.02504 seconds w
Press any key to continue . . .
```



```
Hello 3A-OOP [2020]
```

File I/O Example: Writing

First Method (use the constructor)

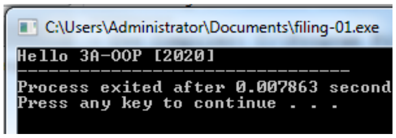
```
#include <fstream>
using namespace std;
int main()
{/* declare and automatically open the file*/
ofstream outFile("fout.txt");
outFile << "Hello World!";
outFile.close();
return 0;
}
```

Second Method (use Open function)

```
#include <fstream>
using namespace std;
int main(){// declare output file variable
ofstream outFile;
// open an exist file fout.txt
  outFile.open("fout.txt");
//behave just like cout, put the word into  the file
outFile << "Hello World!";
outFile.close();
return 0;}
```

Reading File- Example

```
int main()
{
//Declare and open a text file
ifstream openFile("new_file.txt");
string line;
while(!openFile.eof())
{
        //fetch line from data.txt and put it in a string
        getline(openFile, line);
        cout << line;
}
openFile.close(); // close the file
   return 0; }
```



```
Hello 3A-OOP [2020]
--------------------------------
Process exited after 0.007863 second
Press any key to continue . . .
```

File I/O Example: Reading

Read char by char

```
#include <iostream>
#include <fstream>
int main()
{//Declare and open a text file
ifstream openFile(“data.txt");
 char ch;
 //do until the end of file
while( ! OpenFile.eof() )
{
OpenFile.get(ch); // get one character
cout << ch;   // display the character
}
OpenFile.close(); // close the file
    return 0;
}
```

Read a line

```
#include <iostream>
#include <fstream>
#include <string>
int main()
{//Declare and open a text file
ifstream openFile("data.txt");
string line;
while(!openFile.eof())
{//fetch line from data.txt and put it in a string
getline(openFile, line);
cout << line;
}
openFile.close(); // close the file
    return 0; }
```

File Open Mode

```
#include <fstream>
int main(void)
{
ofstream outFile("file1.txt", ios::out);
outFile << "That's new!\n";
outFile.close();
         Return 0;
}
```

If you want to set more than one open mode, just use the **OR** operator- **|**. This way:

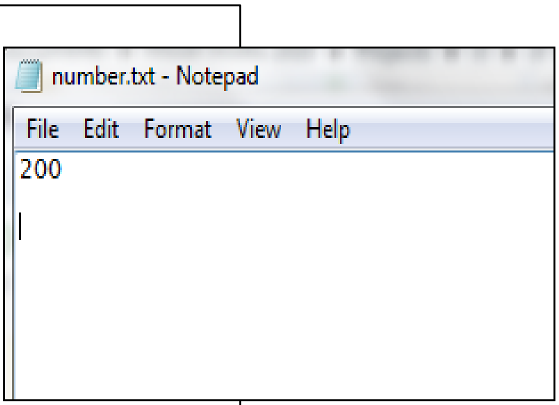
ios::ate | ios::binary

File format

- In c++ files we (read from/ write to) them as a stream of characters
- What if I want to write or read numbers ***?***

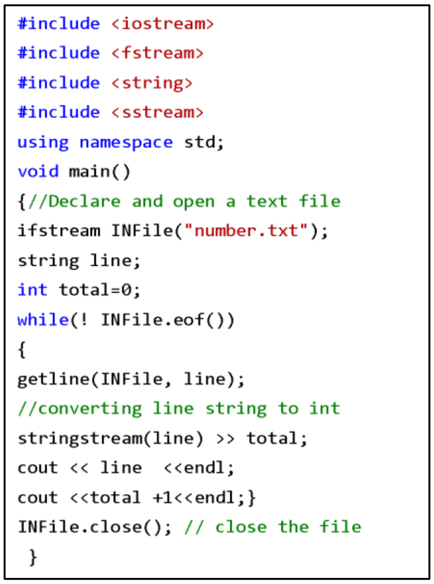
Example writing to file

```
#include <iostream>
#include <fstream>
using namespace std;
void  main()
{
ofstream outFile;
// open an exist file fout.txt
    outFile.open("number.txt",ios::app);

if (!outFile.is_open())
{ cout << " problem with opening the file ";}
else
{outFile <<200 <<endl ;
cout << "done writing" <<endl;}

outFile.close();

}
```

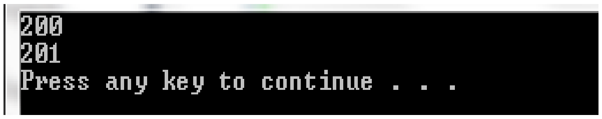


200

Example Reading from file

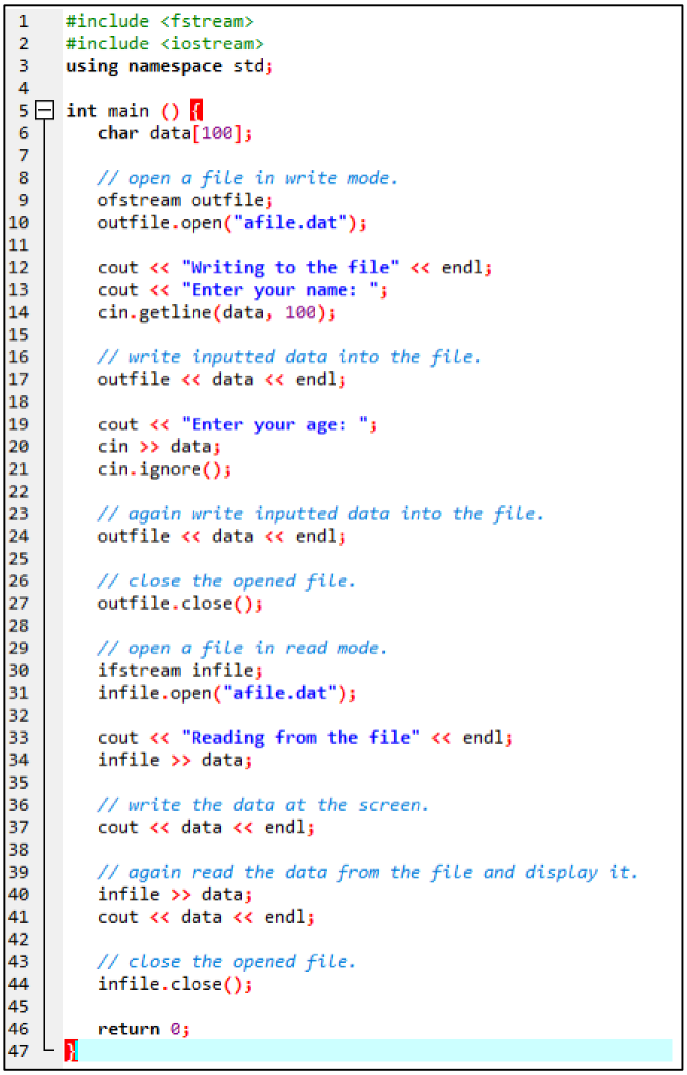


```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
using namespace std;
void main()
{//Declare and open a text file
ifstream INFile("number.txt");
string line;
int total=0;
while(! INFile.eof())
{
getline(INFile, line);
//converting line string to int
stringstream(line) >> total;
cout << line  <<endl;
cout <<total +1<<endl;}
INFile.close(); // close the file
 }
```

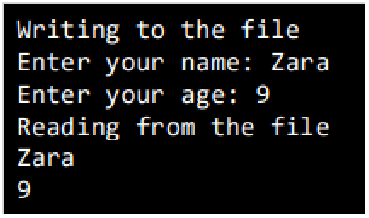


```
200
201
Press any key to continue . . .
```

Example



```
#include <fstream>
#include <iostream>
using namespace std;

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 data into the file.
   outfile << data << endl;

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

   // again write inputted data into the file.
   outfile << data << endl;

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

   // open a file in read mode.
   ifstream infile;
   infile.open("afile.dat");

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

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

   // again read the data from the file and display it.
   infile >> data;
   cout << data << endl;

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

   return 0;
}
```

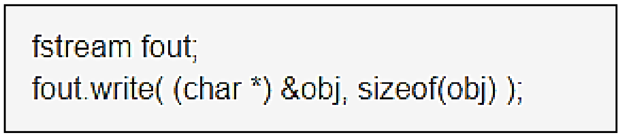


```
Writing to the file
Enter your name: Zara
Enter your age: 9
Reading from the file
Zara
9
```

Examples make use of additional functions from cin object, like getline() function to read the line from outside and ignore() function to ignore the extra characters left by previous read statement

File-handling using OOP concept

Syntax of write () function

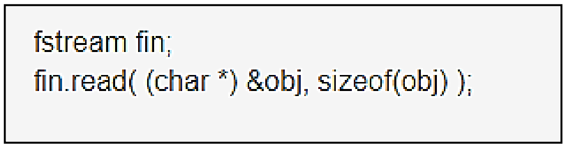


```
fstream fout;
fout.write( (char *) &obj, sizeof(obj) );
```

The write() function takes two arguments.
&obj : Initial byte of an object stored in memory.
sizeof(obj) : size of object represents the total number of bytes to be written from initial byte.

File-handling using OOP concept

Syntax of read () function



```
fstream fin;
fin.read( (char *) &obj, sizeof(obj) );
```

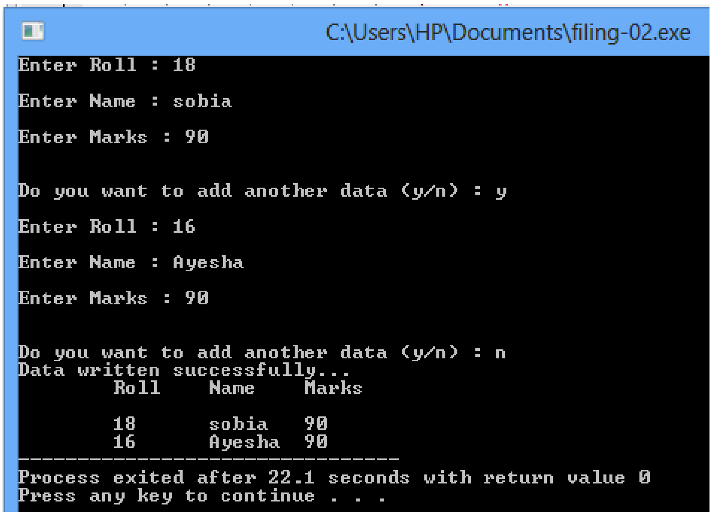
The read() function takes two arguments.
&obj : Initial byte of an object stored in file.
sizeof(obj) : size of object represents the total number of bytes to be read from initial byte.

Program as Example

```
#include <iostream>
#include <fstream>
#include <conio.h>
using namespace std;
    class Student      {
        int roll;
        char name[25];
        float marks;
        void getdata()
        {
            cout<<"\n\nEnter Roll : ";
            cin>>roll;
            cout<<"\nEnter Name : ";
            cin>>name;
            cout<<"\nEnter Marks : ";
            cin>>marks;
        }
    Void    fetchdata()                    {
cout<<"\n\t"<<roll<<"\t"<<name<<"\t"<<m
arks;}
        public:
void AddRecord()
        {
            fstream f;
            Student Stu;

f.open("Student.txt",ios::app|ios::binary);
            Stu.getdata();
            f.write(    (char    *)    &Stu,
sizeof(Stu) );
```

```
                f.close();
            }
            void Display()
            {
                fstream f;
                Student Stu;

f.open("Student.txt",ios::in|ios::binary);

cout<<"\n\tRoll\tName\tMarks\n";
                while(
(f.read((char*)&Stu,sizeof(Stu))) != NULL )
                            Stu.fetchdata();
                f.close();
            }};
    int  main()
      {
            Student S;
            char ch='n';
            do
            {
                S.AddRecord();
                cout<<"\n\nDo you want to add
another data (y/n) : ";
                ch = getche();
            } while(ch=='y' || ch=='Y');
            cout<<"\nData                written
successfully...";

            S.Display(); }
```

Output:



```
Enter Roll : 18

Enter Name : sobia

Enter Marks : 90


Do you want to add another data (y/n) : y

Enter Roll : 16

Enter Name : Ayesha

Enter Marks : 90


Do you want to add another data (y/n) : n
Data written successfully...
        Roll    Name    Marks

        18      sobia   90
        16      Ayesha  90
--------------------------------
Process exited after 22.1 seconds with return value 0
Press any key to continue . . .
```

Activity

1. **QUESTION#1**

   Write a program to implement I/O operations on characters. I/O operations includes inputting a string, calculating length of the string, Storing the String in a file and fetch the stored characters from it.

2. **QUESTION#2**

   Write a program to copy the contents of one file to another

3. **QUESTION#3**

   Write a function in C++ to print the count of word the as an independent word in a text file STORY.TXT.for example, if the content of the file STORY.TXT is

   There was a monkey in the zoo. The monkey was very naughty.

   Then the output of the program should be 2

4. **QUESTION#4**

   Take a class Person having two attributes name and age.

   Include a parameterized constructor to give values to all data members.

   In main function
   i. Create an instance of the person class and name it person1.
   ii. Create a binary file person.bin and write person1 object into it.
   iii. Read the person1 object from the file.
   iv. Return 0

5. **QUESTION#5**

   Take a class Participant having three attributes (ID, name and score) and following member functions

   - Input () function takes data of the object and stores it in a file name participant.dat

- Output () function takes id from user and show respective data of that id.
- Max () gives the highest score of the Participant in the file

6. **QUESTION#6**

   Write a function in C++ to count and display the number of lines not starting with alphabet 'A' present in a text file "STORY.TXT".

   Example:

   1. If the file "STORY.TXT" contains the following lines,
   2. The rose is red.
   3. A girl is playing there.
   4. There is a playground.
   5. An aeroplane is in the sky.
   6. Numbers are not allowed in the password.
   7. The function should display the output as 3.