File operations

Summarize

Q1 List all the stream class in stream operation.

→ * File handling techniques in C++ support file manipulation in the form of stream object
* Stream object in c++ are cin and cout.
* They are used to Deal with standard i/p & o/p devices.
* C++ class supports hierarchy of classes that are used to manipulate both console and disk files called stream classes.

Classes

fb
of
if
fstream
fstream base.

- filebuf – set the file buffers to read and write contains close(), open() members.
- fstreambase – basic class of fstream, ifstream & ofstream contain close(), open()
- ofstream – continuously o/p operations contain open() with default o/p mode in case already file exists
- ifstream – open() & close with default input mode
- ofstream – inherits get(), getline(), read(), seekg() & tellg() from ifstream
- fstream – continuously i/p & o/p operations. contains open() with default input mode. inherits all fun from istream & ostream class through iostream

rd but()

---

fclose

← Sequence of characters that are transfer between the program and I/P/O/P devices.

* the fclose function() closes a stream
* This fun^n deletes all the Buffers that are associated with the stream before closing it.
* If it not called explicitly the O.S will normally close file when program terminates.

Scanned with CamScanner

Syntax stream_object.open("filemode", openg mo

ios::in

open file for reading only.

This mode should be specified for I/P file

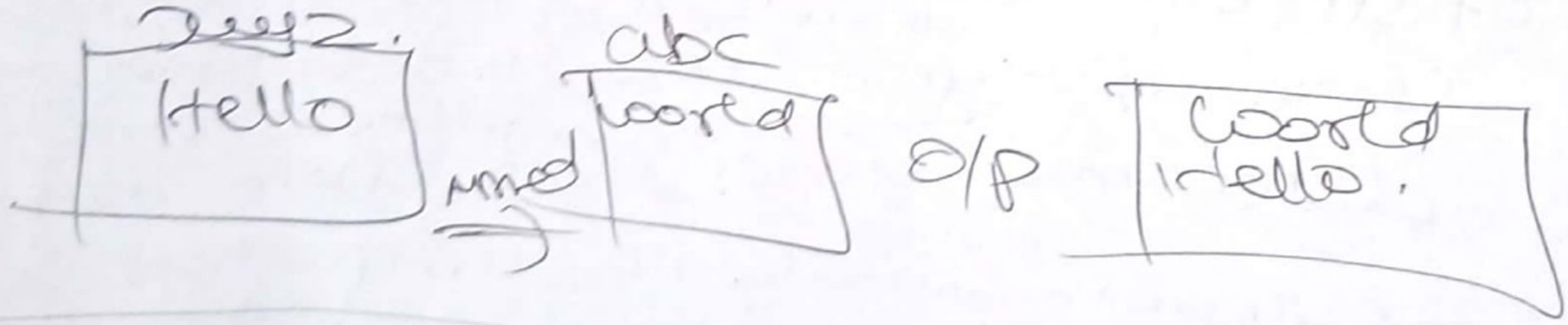
ios::out

Openfile for writing only.

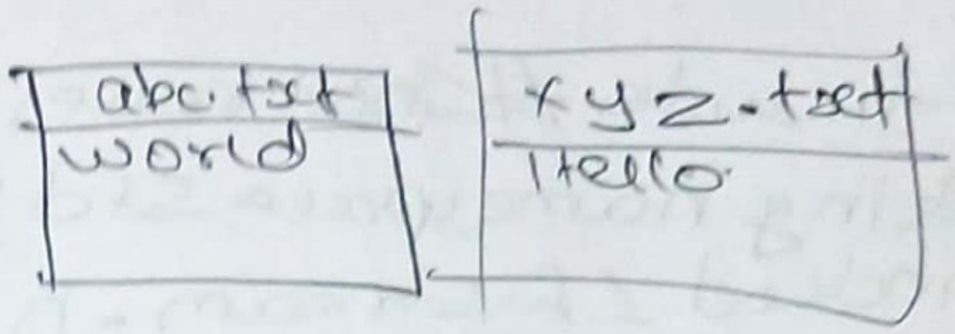
This mode should be specified for O/P files.

ios::app

append to end of file or
write occurs at end of file set a
pointer to end of file.

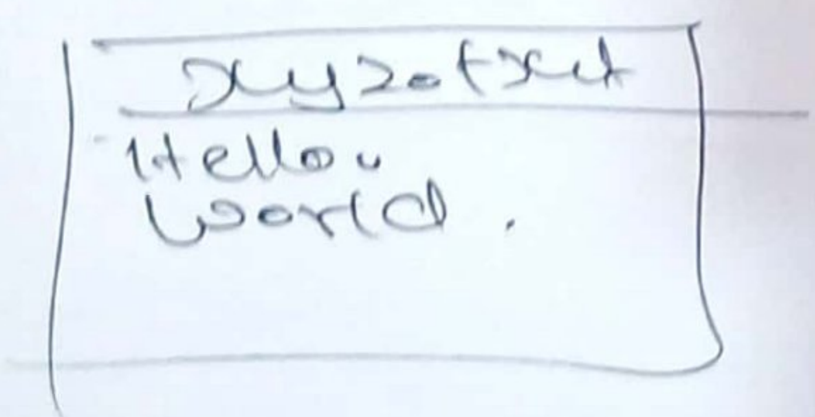


t program to append data from abc.txt to xyz.txt.




```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
void main()
{
ifstream fin;   //ip operation
fin.open("abc.txt", ios::in);
                //read.
ofstream fout;  //op operation.
fout.open("xyz.txt", ios::app);
                //app.
if(fin)
{
cout<<"File not found";
}
else
{
fout<<fin.rdbuf(); //read buffer
}
fin.close();
fout.close();
return 0;
}
```

So



Scanned with CamScanner

* Count no of lines in files spaces.

```
#include <iostream>
using namespace std;
#include <fstream.h>
void main()
{
  ifstream X;
  char ch;
  int n=0;
  X.open("abc.txt");
    while(!X)
    if (!x)
      {
      cout<<"file not found";
        }
    else
    {
      while(x)
      {
       X.get(ch);
      if(ch==' ') of if(ch==',')
        {
        ntt;
        }
      }
cout<<"No. of Spaces"<<n;
for X.close();
}
```

no. of ~~spaces~~ lines in file.

```
#include <iostream.h>
#include <fstream.h>

void main()
{
  ifstream f;
  char ch;
  int n=0;
  f.open("abc.txt");
  if(!f)
    {
    cout<<"file not found";
    }
else
  {
    while(f)
      {
      f.get(ch);
      if(ch=='\n')
        {
        n++;
        }
      }
  cout<<"No. of lines are "<<n;
  f.close();
  }
```

O/P

No. of lines are : 3

detection of end of file.

```
#include <iostream>
#include <fstream>
int main()
{
ifstream fi;
fi.open("xyz.txt");
char ch;
if(fi!='\0')
  {while(!fi.eof())
    {
     fi.get(ch);
    cout<<ch;
    }
    fi.close();
  }
else
 {
 cout<<"detection end of file";
 }
```


Scanned with CamScanner

Write P to write 'welcome to poly' in f
Then read the data from file and d
9t on screen.

```
#include<iostream>
#include<fstream.h>
Using namespace std;
void main().
{
char str[50]="welcome to poly";
ofstream f;
 f.open("output.txt");
  f>>str;
  f.close();
 ifstream fin;
 fin.open("output.txt");
   while(!fin.eof())
    {
    fin.getline(str,50);
    cout<<str;
   }
fin.close()
}
```


Scanned with CamScanner

file mode

read()

write()

open()

close()

Scanned with CamScanner

04/11/23

Unit 5: Files operation (coverage - 4)

Q) List all the classes in stream operations.

⇒

- File handling techniques of C++ support file manipulation in the form of stream objects
- The stream object in c++ are cin & cout.
- Cin & cout are used to deal with Standard input & output devices.
- The C++ input output system support hierarchy of object classes, these are used to manipulate both console and disk files called stream clads.

List stream classes

	Name	Des
①	filebuf	It purpose is to set the file buffer to read and write contains close() open() as a member.
②	fstream base	it the base class of fstream, ifstream & ofstream contains close() & open().
③	ifstream (input operation)	it provides input operations. contains open() & close() with default input mode. inherits the functions get(), getline(), read(), seekg() & tellg(), drop from istream.

Scanned with CamScanner

(2) Scope resolution operator

(4) iostream	It provides output operations. contains open() with default output mode. inherits put() , seekp() , tellp() & write() from ostream.
(5) fstream	It allows simultaneous input & output operations. contains open() with default input mode. inherits all the function from istream & ostream classes through iostream.

file modes

Q// List / Explain file mode used to perform file operations.

modes	

We can open files in C++ in following two ways
- By using constructor
- Using function open()

C++ provides a mechanism of opening a file in different modes.

Syntax
stream_object.open("filename", openmode);

Here the open function takes two arguments
1. filename
2. opening mode.

ios::in
ios::out are default values.

ios::in	Open file for reading only. This mode should be specified for input files.
ios::out	Open file for writing only. This mode should be specified for output files.
ios::binary	Open file in binary mode. This mode should be specified for large databases.
ios::trunc	Delete contents of file delete if exist.
ios::nocreate	open file if file does not exists. It is relevant directories.
ios::noreplace	Open file if file already exits. It will allow us to open file with same name.
ios::ate	Goto to end of file. It will set to end of file it will also allow to add or modify