Data File Handling

Date

Programming with C++

Introduction : Files are used to store information. File is a bunch of bytes stored on some storage device like disk.

In C++ file I/O are implemented through a header file called <fstream.h>

Various data types :

(i) ~~i~~fstream (ii) ifstream (iii) ofstream

(1) fstream : File can be used for both input & output.

(2) ifstream : File can be used for i/p purpose

(3) ofstream : File can be used for o/p

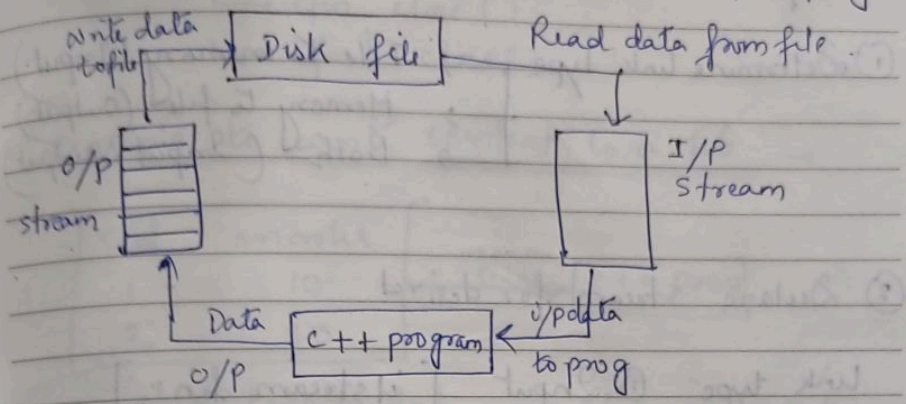
Stream : It is a sequence of bytes.

(1) Input stream : Supplies data to the program.

REDMI NOTE 9 PRO
AI QUAD CAMERA
2022/9/2 11:13

Date

(2) Output stream : Receives data from program.



Difference between text and Binary file.

Text file	Binary file
(i) Stores information in ASCII char.	(i) In same format as held in memory.
(ii) Each line is terminated by special char called EOL.	(ii) No such delimeter.
(iii) Internal translations take place.	(iii) No translation.
(iv) Slow to operate	(iv) Fast.

2022/9/2 11:13
REDMI NOTE 9 PRO
AI QUAD CAMERA

Date

Steps to Process files in the C++ Program:

(1) Determine link type
- → File to memory (Input)
- → Memory to file (Output)
- → Both (Input & Output)

(2) Declare stream for desired link type
- (i) Input [ifstream fin;]
- (ii) Output [ofstream fout;]
- (iii) Both [fstream finout;]

(3) Attach desired stream with the file:
ifstream fin ("Abc.txt", ios::in)
ofstream fout ("cba.txt", ios::out)
fstream finout ("xy.txt", ios::in|ios::out);

(4) Process as required.
- ↳ Read from file
- ↳ write to file.

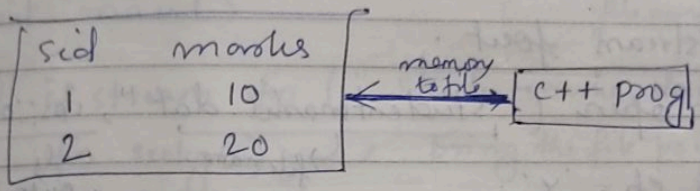
2022/9/2, 11:13
REDMI NOTE 9 PRO
AI QUAD CAMERA

2022/9/2 11:13

Date

(5) close file : link with stream
↳ fin.close();
↳ fout.close();

example to write content to a file : →



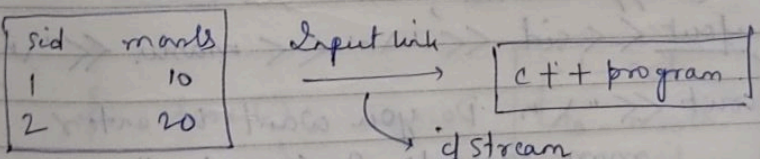
student marks.dat

ofstream is to be used.

program

Bind the file to ofstream.
↳ write to file
↳ close link.

example to read from file :



Student marks.dat

ifstream
↳ Bind file
↳ Reading content from file
↳ link close.

program

REDMI NOTE 9 PRO
AI QUAD CAMERA

Date

(Program: ① Write data to file using ofstream class)

```
#include <iostream.h>
#include <fstream.h>
int main()
{ ofstream fout;
  fout.open ("studentmarks.dat", ios::out);
                  ↳ file name.        ↳ output mode
  char ch = 'Y';
  int sid, marks;
  while (ch == 'y' || ch == 'Y')
  { cout << "Enter student id:\n";
    cin >> sid;
    cout << "Enter student marks:\n";
    cin >> marks;
    fout << sid << '\n' << marks << '\n';
    cout << "\n Do you want to enter
         more records ?. (Y/N): ";
    cin >> ch;
  }
  fout.close(); ⟶ close the link.
  return 0;
}
```

↙ is used to write on file

2022/9/2 13:13
REDMI NOTE 9 PRO
AI QUAD CAMERA

Date

Program 2: To Read data from file using ifstream class.

```
#include <iostream.h>
#include <fstream.h>
int main()
{ ifstream fin ("studentmarks.dat", ios::in);
  fin.seekg(0);  // bring the file pointer at the beginning.
  int sid, marks;
  for (int i=0; i<2; i++)
  { fin >> sid;
    fin >> marks;
    cout << "student id is: " << sid;
    cout << "\t marks: " << marks "\n";
  }
  fin.close();
  return 0;
}
```

filename, mode

1- sid
1a - marks
2 - sid
2a - marks

close the link.

2022/9/2 11:13
REDMI NOTE 9 PRO
AI QUAD CAMERA

2022/9/2 11:17

Date

Program 3: to append data file.

```
#include <iostream.h>
#include <fstream.h>
int main()
{ ofstream fout;
  fout.open("studentmarks.dat", ios::app);
  char ch='y';
  int sid, marks;
  while (ch='y' || ch='Y')
  { cout << "Enter student id:\n";
    cin >>sid;
    cout << "Enter student marks \n";
    cin >> marks;
    fout <<sid <<'\n' <<marks <<'\n';
    cout <<"\n Do you want to enter
    more records?(Y/N)";
    cin >>ch;
  }
fout.close();
return 0;
}
```

REDMI NOTE 9 PRO
AI QUAD CAMERA

Date

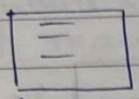
example of read & write : file1 → file2


```
ifstream fin("file1.txt", ios::in);
ofstream fout("file2.txt", ios::out);
while (!fin.eof())
{
    char c = fin.get();
    fout << c;
}
```

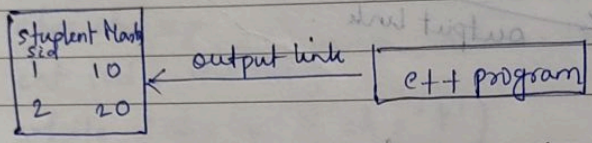
REDMI NOTE 9 PRO
AI QUAD CAMERA
2022/9/2 11:14

Date

example to append the file.
↳ Retaining the old data.



f1.txt → out mode → old data will get removed.



student marks.dat [mode will be append]
↓
to add more data

```
ofstream fout;
fout.open("studentmarks.dat", ios::app);
---
---
---
{
fout << sid;
}
```

Program is 3.

2022/9/2 11:14
REDMI NOTE 9 PRO
AI QUAD CAMERA

2022/9/2 11:14

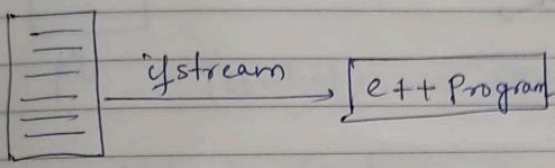
Date

Sequential I/O with files:

example of using get() function:

get() functn reads single character from associated stream. It is only capable of handling single char. at a time.

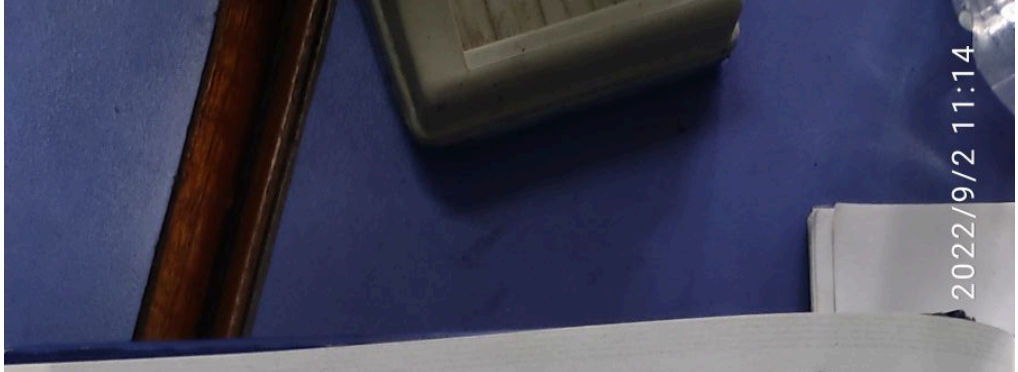
↳ byte. oriented → Read single byte of data.




```
ifstream fin;
fin.open("file1.txt", ios::in);
{ fin.get(ch);
}
```

```
#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
int main()
{ system("cls");
```

REDMI NOTE 9 PRO
AI QUAD CAMERA



Date

```
char ch;
ifstream fin;
fin.open("studentmarks.dat", ios::in);
while(fin)  → true till last char. of the file.
  { fin.get(ch);
    cout << ch;
  }
fin.close();
return 0;
}
```

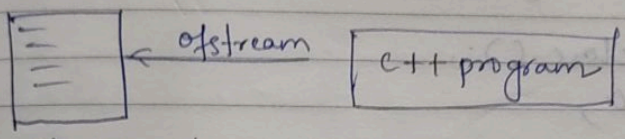
2022/9/2 11:14
REDMI NOTE 9 PRO
AI QUAD CAMERA

2022/9/2 11:14

Date

example of using put() function:

Put() funct^n also handles single char.
Put () funct^n will write a byte of data.
→ byte oriented → write single byte of data.



file (content) ⟶ "EEC classes"

```
ofstream fout;
fout.open("file.txt",ios::app);
fout.put((content));
```

program

```
#include<iostream.h>
#include<fstream.h>
#include<stdlib.h>
int main()
{ char content;
  system("cls");
```

REDMI NOTE 9 PRO
AI QUAD CAMERA

Date

```
ofstream fout;
fout.open("newfile.txt", ios::app);
cout<<"enter the content for file\n";
cin.get(content);
fout.put(content);
fout.close();
return 0;
}
```

—x—

Example of using getline() function:

getline() function reads character from i/p stream. Difference from get() function is that getline() function reads and removes the delimiter new-line character from the i/p stream.

```
ifstream fin( );
fin.getline(ch, 50, ':');
```

(ch → char(delimeter), int(no. of characters))

2022/9/2 11:14
REDMI NOTE 9 PRO
AI QUAD CAMERA

2022/9/2 11:12
REDMI NOTE 9 PRO
AI QUAD CAMERA

Date

Exception Handling in C++.

⇓

error.

The process of converting system error message into user friendly error message is known as Exception Handling.

It is an event, which occurs during the execution of program, that disrupts the normal flow of the program instruction.

Errors can be classified into 2 types.

① Compile time error ⇒ error caught during compile time. It includes library reference, syntax error of incorrect class input

② Run time error → Also known as exception. An exception caught during runtime creates serious issues.

2022/9/2 11:12
REDMI NOTE 9 PRO
AI QUAD CAMERA

Date

eg. user divides a no. of zero, this will compiles successfully but an exception of runtime error will occur due to which our application will be crashed.

⇒ In order to avoid this we will introduce exception handling technique in our code.

⇒ Exception Handling Mechanism:

1. Find problem (Hit the exception)
2. Inform about its occurence (throw the exception)
3. Receive error information (catch the exception)
4. Take proper action. (Handle the exception)

Object oriented Programming

Date

c++ fully supports object-oriented programming, including the four pillars of object oriented development -

- Encapsulation
- Data hiding
- Inheritance
- Polymorphism.

REDMI NOTE 9 PRO
AI QUAD CAMERA
2022/9/2 11:12