



UNIVERSITÀ
degli STUDI
di CATANIA

File IO in C++

Corso di programmazione I AA 2019/20

Corso di Laurea Triennale in Informatica

Prof. Giovanni Maria Farinella

Web: <http://www.dmi.unict.it/farinella>

Email: gfarinella@dm.unict.it

Dipartimento di Matematica e Informatica

Classi ofstream, ifstream ed fstream

Classi di Input/Output a caratteri con i file:

- ofstream scrittura
- ifstream lettura
- fstream lettura e scrittura

```
#include <iostream>  
#include <fstream> ←
```

```
std::fstream  
std::ifstream  
std::ofstream
```

Apertura di un file

Aprire un file in lettura con ifstream (costruttore):

```
1  #include <fstream>
2  #include <iostream>
3
4  std::ifstream myfile("test.txt");
5  if (!myfile.is_open())
6      cerr << "Errore apertura file!" << endl;
```

Apertura di un file

Aprire un file in **lettura con ifstream** (metodo open()):

```
1  #include <fstream>
2  #include <iostream>
3
4  std::ifstream myfile(); //NB: costr. default
5  → myfile.open("test.txt"); //NB: metodo open
6  → if(!myfile.is_open())
7      cerr << "Errore apertura file!" << endl;
```

Apertura di un file

Aprire un file in **scrittura con ofstream**.

```
1 #include <fstream>
2 #include <iostream>
3
4 std::ofstream myfile(); //NB: costr. default
5 ➔ myfile.open("test.txt"); //NB: metodo open
6 if(!myfile.is_open())
7     cerr << "Errore apertura file!" << endl;
```

L'interfaccia è identica a quella di `ifstream`, per l'apertura si può usare il metodo `open()` oppure il costruttore.

Apertura di un file

Aprire un file in lettura e/o scrittura con fstream:

```
1 #include <fstream>
2 #include <iostream>
3 std::fstream myfile(); //NB: costr. default
4 myfile.open("test.txt", std::fstream::out);
5 if (!myfile.is_open())
6     cerr << "Errore apertura file!" << endl;
```

NB: Per `fstream` modalità di apertura file va specificata!

Apertura di un file

Apertura file in lettura/scrittura:

```
1 #include <fstream>
2 #include <iostream>
3 std::fstream myfile(); //NB: costr. default
4 myfile.open("test.txt", std::fstream::out | std::fstream::in);
5 if (!myfile.is_open())
6     cerr << "Errore apertura file!" << endl;
```

NB: Per **combinare apertura e scrittura**, si usa **operatore bitwise '|'** (bitwise OR).

Gli operatori bitwise del C++ si usano per fare operazioni logiche sui singoli bit contenuti nelle aree di memoria (ad esempio una variabile intera).

https://en.cppreference.com/w/c/language/operator_arithmetic

Lettura e scrittura di caratteri da/su file

La **scrittura** ed il successivo **controllo dello errore** può avvenire mediante i soliti operatori e/o i metodi usati con gli stream

i) scrittura byte su file: operatore di inserimento '<<'

ii) eventuale controllo errore: metodo `std::basic_ios.fail()`

```
1 #include <fstream>
2 #include <iostream>
3 fstream myfile("test.txt", std::fstream::out);
4 myfile << "pippo test";
5 if(myfile.fail())
6     cerr << "Error writing on file.." << endl;
```


Lettura e scrittura di caratteri da/su file

La **scrittura** ed eventuale **controllo errore** può avvenire anche mediante

i) operatore di inserimento << + operatore '!''

```
1  #include <fstream>
2  #include <iostream>
3  fstream myfile("test.txt", std::fstream::out);
→ 4  if(!(myfile << "pippo test"))
5      cerr << "Error writing on file.." << endl;
```

Lettura e scrittura di caratteri da/su file

Lettura ed eventuale **controllo errore** può avvenire mediante:

i) operatore di estrazione '>>' ii) il metodo

`std::basic_ios.fail()`

```
1  #include <fstream>
2  #include <iostream>
3  fstream myfile("test.txt", std::fstream::in);
4  string s; ←
5  myfile >> s;
6  if (myfile.fail()) ←
7      cerr << "Error reading from file.." << endl;
```

Lettura e scrittura di caratteri da/su file

Lettura ed eventuale **controllo errore** può avvenire mediante operatore di estrazione '>>' + operatore '!'

```
1  #include <fstream>
2  #include <iostream>
3  fstream myfile("test.txt", std::fstream::in);
4  string s;
5  if(!(myfile >> s))
6      cerr << "Error reading from file.." << endl;
```

È buona norma **chiudere** il descrittore associato al file dopo le operazioni di lettura/scrittura.

Metodo `close()` di `std::fstream`.

```
1  #include <fstream>
2  #include <iostream>
3  fstream myfile("test.txt", std::fstream::out);
4  if (!(myfile << "test"))
5      cerr << "Error writing on file.." << endl;
6  myfile.close();
```

~~A14_01_ifstream.cpp~~

~~A14_02_ifstream.cpp~~

~~A14_03_ofstream.cpp~~

~~A14_04_ofstream.cpp~~

~~A14_05_fstream_read_numbers_fromfile.cpp~~

~~A14_05_fstream_write.cpp~~

~~A14_06_fstream_write.cpp~~

~~A14_06_fstream_write_numbers_onfile.cpp~~

~~A14_07_fstream_read.cpp~~

~~A14_08_fstream_read.cpp~~

FINE