



Facultatea de Inginerie Electrică



UNIVERSITATEA TEHNICĂ  
DIN CLUJ-NAPOCA



# PCLP 2

## Programarea calculatoarelor si limbaje de programare 2

PCLP2

An I semestrul II



*"Coding is easy when you C it in action."*

# Cap. 11

## Operatii cu fisiere in C++

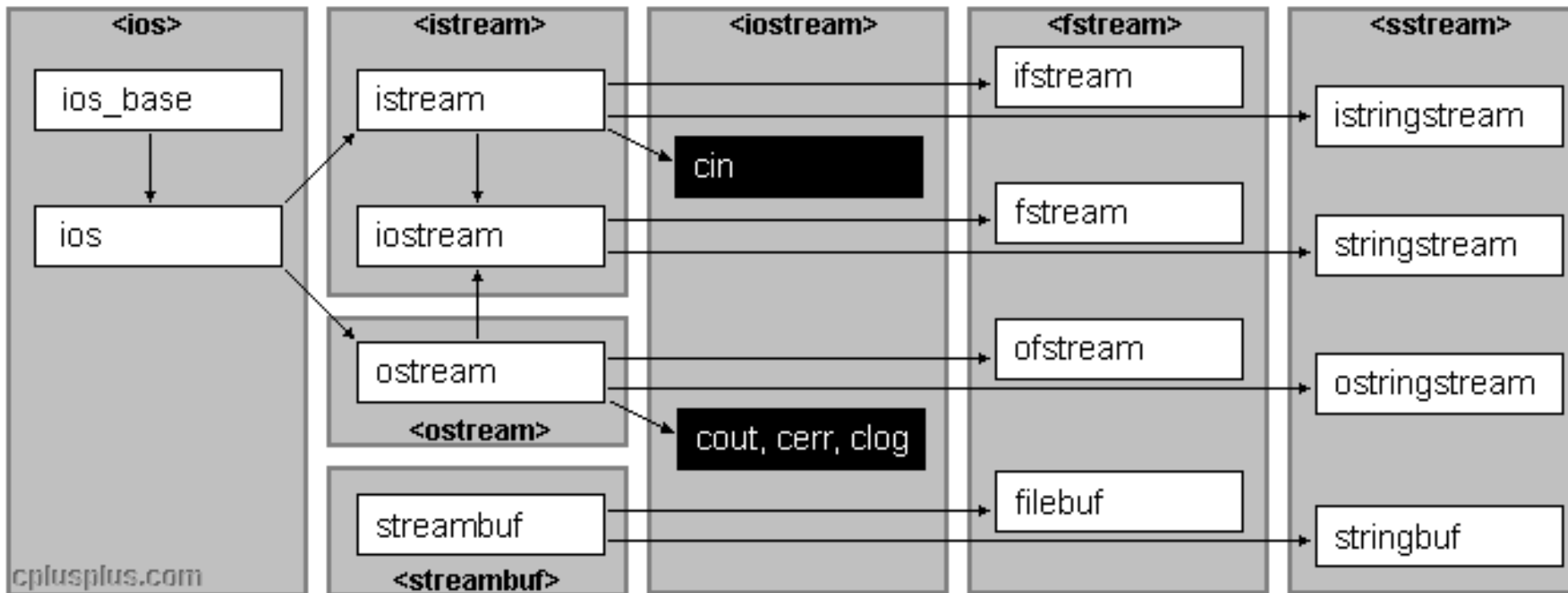
---

**11.1** Operatii cu fisiere in C++

**11.2** Exemple

# 11.1 Operatii cu fisiere in C++

## Ierarhia de clase ios: IOStream



# 11.1 Operatii cu fisiere in C++

## Fluxuri-stream

### DEFINIRE

Intrările și ieșirile în limbajul C++ sunt implementate cu ajutorul **fluxurilor (stream)**.

**Fluxurile** sunt **obiecte** care transportă și formatează șiruri de bytes.

**Fluxurile** pot fi:

- unidirecționale** (doar de intrare –**ifstream** sau doar de ieșire-**ofstream**)
- bidirecționale** (permit și intrări și ieșiri-**fstream**).

Clasele care reprezintă fluxuri asociate fișierelor sunt definite în **biblioteca <fstream>**.

Aceste clase sunt:

- ifstream**: flux unidirecțional care permite **citirea** de date dintr-un fișier
- ofstream**: flux unidirecțional care permite **scrierea** de date într-un fișier
- fstream**: flux bidirecțional care permite **citiri și scrieri** în/din fișier

# 11.1 Operatii cu fisiere in C++

## Fluxuri-stream

### DEFINIRE

#### optiuni

#### Deschiderea fișierelor in C++

1. prin **constructor** sau
2. prin apelarea **funcției membru open()**.

Ambele variante primesc ca parametri numele fișierului și opțiunile de deschidere:

#### 1.Deschidere fisier prin constructor:

```
fstream fisier("nume_fisier", optiuni);  
ofstream fisierout("nou.txt");  
ifstream fisierin("nou.txt");
```

#### 2.Deschidere fisier cu funcția open:

```
fstream fisier;  
fisier.open("nume_fisier", optiuni);
```

Constantă	Efect
<code>ios::in</code>	Deschide un fișier pentru citire. Opțiunea este folosită și pentru a evita suprascrierea fișierului în cazul în care acesta există.
<code>ios::out</code>	Deschide un fișier pentru scriere.
<code>ios::app</code>	Deschide fișierul pentru adăugare. Toate scrierile se vor face la sfârșitul fișierului.
<code>ios::ate</code>	Deschide fișierul și se poziționează la sfârșit.
<code>ios::trunc</code>	Șterge conținutul fișierului la deschidere (dacă acesta există).
<code>ios::binary</code>	Deschide fișierul în mod binar (implicit este text).

#### optiuni implicite:

<code>ifstream</code>	<code>ios::in</code>
<code>ofstream</code>	<code>ios::out</code>
<code>fstream</code>	<code>ios::in   ios::out</code>

Verificarea deschiderii fișierului se poate face folosind funcția membru: `is_open()`. Implicit fisierele se considera de tip `.txt`

# 11.1 Operatii cu fisiere in C++

## Deschiderea fisierelelor pentru scriere: **ofstream()**

### SINTAXA

**Format :** `ofstream(const char *nume_fisier, int mod_acces=ios::out);`

**Obs:** trebuie inclusa biblioteca `<fstream>`

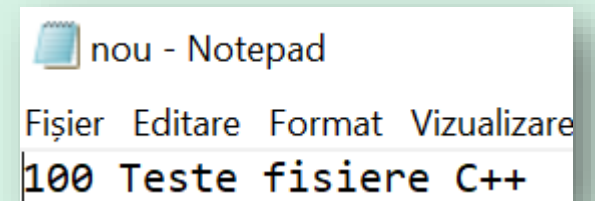
Pentru a asocia un **flux de ieşire** (stream) unui fisier, este necesară **crearea unui obiect** de tipul **ofstream**.

Clasa **ofstream** este *derivată* din clasele **ostream** și **fstream**

### EXEMPLU

**Ex. :** Scrie intr-un fisier o variabila intreaga si un text. Se asociază fluxului de ieşire (obiectului `fisierout`) fişierul `text.txt`

```
#include <iostream> //pentru cout
#include <fstream> //pentru ofstream, close
using namespace std;
int main()
{ int d=100;
  ofstream fisierout("nou.txt");// Deschide fisierul text nou.txt pentru scriere
  fisierout << d; //scrie valoarea variabilei d in fisier
  fisierout << " Teste fisiere C++"; //Scrie un text in fisier
  fisierout.close(); // Inchide fisierul.
  cout << "Fisierul a fost creat.";return 0;}
```



# 11.1 Operatii cu fisiere in C++

Deschiderea unui fisier pentru citire: **ifstream()**

## SINTAXA

Format: **ifstream(const char \*nume\_fisier, int mod\_acces=ios::in);**

Obs: trebuie inclusa biblioteca <fstream>

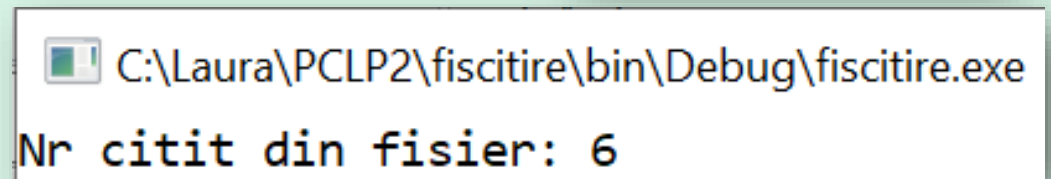
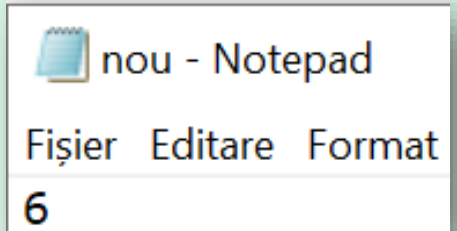
- ❑ Pentru a asocia un **flux de intrare** unui fisier, este necesară crearea unui obiect de tipul **ifstream**.
- ❑ Clasa **ifstream** este *derivată* din clasele **istream** și **fstream**

## EXEMPLU

Ex. : Citire un intreg din fisierul nou.txt si afisarea lui pe ecran

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{ifstream fisierin("nou.txt"); // Deschide fisierul text nou.txt pentru citire
  int n;
  fisierin >>n; // citeste din fisier un nr intreg si-l atribuie lui n
  cout <<"Nr citit din fisier: "<<n<<endl;
  fisierin.close(); // Inchide fisierul.
  return 0;}

```



# 11.1 Operatii cu fisiere in C++

## Testare erori operatii fisiere in C++

### SINTAXA

Testarea operatiei de deschidere se poate face verificand daca stream-ul este valid asociind obiectului creat pentru stream tipul Boolean: **true** daca **operatia se face cu success** , **false** daca apar **erori**.

```
// deschidere fisier
```

```
fstream fisier("studenti.txt", ios::in);
```

```
// verificare operatie deschidere fisier
```

```
if (!fisier) {cout << "Fisierul nu poate fi deschis!" << endl; }
```

Testarea se poate face si cu functia **is\_open()** care returneaza true daca fisierul este asociat stream-ului obiect deschis si false , sau cu functia **fail()**

```
ofstream test("text.txt");
```

```
if (!test.is_open()) cout<<"Fisierul nu poate fi deschis!"<<endl;
```

```
else cout<<"Fisierul a fost creat. Verificati!"<<endl;
```

```
if (test.fail()) {cout<<"EROARE la deschidere"<<endl;exit(1);} 
```



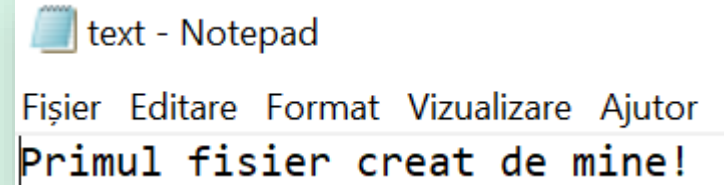
# 11.1 Operatii cu fisiere in C++

## Scriere in fisiere in C++

### EXEMPLU

**Ex. :** Scrierea intr-un fisier in C++ un text din variabila msg[]. Se asociază fluxului de ieşire (obiectului test) fişierul `text.txt`

```
#include <iostream>
#include <string.h> //pentru strlen
#include <fstream>
using namespace std;
int main()
{ofstream test("text.txt");
char msg[]="Primul fisier creat de mine!";
if (!test.is_open())
cout<<"eroare la deschidere fisier!"<<endl;
else {cout<<"Fisierul a fost creat. Verificati!"<<endl;}
test<<msg;test.close();
return 0;
}
```



# 11.1 Operatii cu fisiere in C++

## Testare sfarsit de fisier: eof()

### SINTAXA

Testarea sfarsitului de fisier : functia eof() care returneaza true dacă s-a ajuns la sfarsitul fișierului.

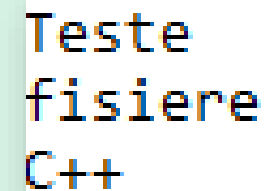
```
ifstream fisierin("nou.txt"); // Deschide fisierul text nou.txt pentru citire
char sir[50];
while (!fisierin.eof())
{ // Citeste cuvant cu cuvant din fisier pana ajunge la sfarsitul fisierului.
  fisierin >> sir;
  //afiseaza fiecare sir pe ecran
  cout << sir << endl;
}
```

# 11.2 Exemple aplicatii cu fisiere in C++

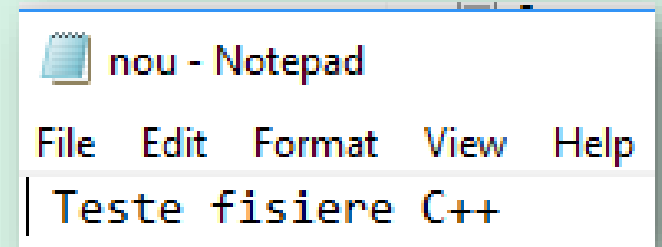
## EXEMPLU

**Ex. :** Se scrie un text intr-un fisier nou.txt , apoi se citesc cuvinte din acest fisier si se afiseaza pe cate o linie noua

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{ ofstream fisierout("nou.txt"); // Deschide fisierul text nou.txt pentru scriere
  fisierout << " Teste fisiere C++"; //Scrie un text in fisier
  fisierout.close(); // Inchide fisierul.
  ifstream fisierin("nou.txt"); // Deschide fisierul text nou.txt pentru citire
  char sir[50];
  while (!fisierin.eof()){ // Citeste cuvant cu cuvant din fisier pana la sfarsit
    fisierin >> sir;    cout << sir << endl; }
  //afiseaza cate un cuvant pe linie pe ecran
  fisierin.close(); // Inchide fisierul.
  return 0;}
```



```
Teste
fisiere
C++
```



# 11.2 Exemple aplicatii cu fisiere in C++

## EXEMPLU

**Ex. :** Programul citește pe rând câte o operație din fișierul de intrare a.txt, sub forma op1 op op2 și se tipărește rezultatul operației în fișierul ab.txt

```
#include <iostream>
#include<fstream>
using namespace std;
int main (void)
{ifstream fis1; ofstream fis2;
int op1,op2,rez; char op;
fis1.open("a.txt"); fis2.open("ab.txt");
if (fis1.fail()) {cout<<"EROARE la deschidere"<<endl;exit(1);}
//fail() testeaza daca se poate deschide fisierul
while (!fis1.eof())
{ fis1>>op1>>op>>op2;
switch (op) {
case '+': rez=op1+op2; break;
case '-': rez=op1-op2; break;
case '*': rez=op1*op2; break;
case '/': if (op2==0) { rez=0;fis2<<"divizor nul, rez="<<rez; } else rez=op1/op2; break;
default: rez=0; fis2<<"operator eronat:";}
fis2<<op1<<op<<op2<<"="<<rez<<endl;} cout<<"Succes"; return 0;}
```

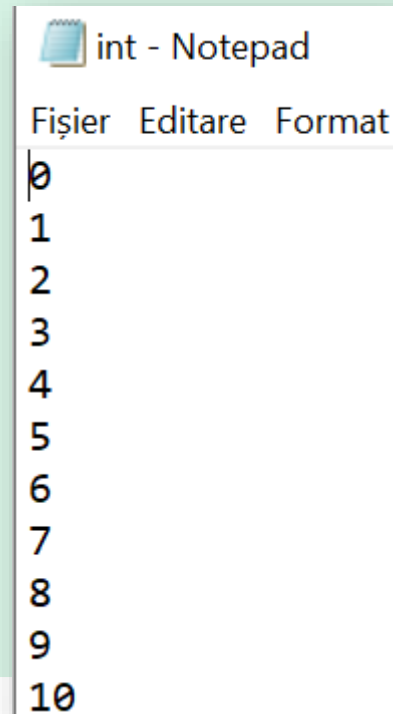
a - Notepad	ab - Notepad
Fișier Editare Format	Fișier Editare Format Vizualizare Ajutor
10+34	10+34=44
560-76	560-76=484
56/7	56/7=8
56/0	divizor nul, rez=056/0=0
6;7	operator eronat:6;7=0

# 11.2 Exemple aplicatii cu fisiere in C++

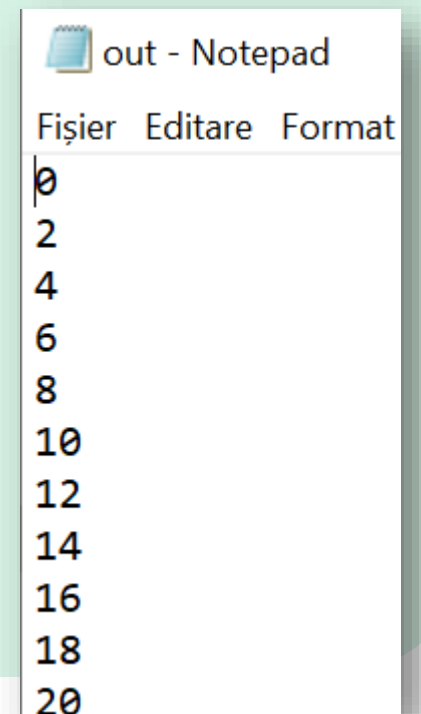
## EXEMPLU

Ex. : Fișierul int.txt se inițializează prin program cu numere de la 0 la 1000 iar dublul acestora numere se tipărește în fișierul out.txt.

```
#include <iostream>
#include <fstream>
using namespace std;
int main(void)
{ofstream fis1, fis2;
int i,j=0,a[1001];
cout<<endl<<"Initalizarea sirului cu nr. de la 0 la 1000";
fis1.open("int.txt"); fis2.open("out.txt");
for (i=0;i<=1000;j++,i++)
{a[i]=j;
fis1<<a[i]<<endl;
fis2<<a[i]*2<<endl;
}
fis1.close();fis2.close();
return 0;}
```



```
int - Notepad
Fișier Editare Format
0
1
2
3
4
5
6
7
8
9
10
```



```
out - Notepad
Fișier Editare Format
0
2
4
6
8
10
12
14
16
18
20
```



# TEST kahoot

Pentru login, introduceti codul afisat pe ecran, in browser la adresa:

<http://kahoot.it>