



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. 7

# Limbaajul C++

---

## I/O in C++

**7. 1 Citirea/afisarea datelor in C++**

**7.2 Utilizarea cin si cout**

**7.3 Citirea/afisarea cu format in C++**

# 7. 1 Citirea/afisarea datelor in C++

## Obiecte standard I/O in C++

### SINTAXA

Cand un program C++ care include biblioteca `iostream.h` este lansat in executie, sunt create si initializate automat **patru obiecte standard**:

- ❑ **cin** gestioneaza intrarea de la intrarea standard (tastatura).
- ❑ **cout** gestioneaza iesirea catre iesirea standard (ecranul).
- ❑ **cerr** gestioneaza iesirea catre dispozitivul standard de eroare (ecranul).
- ❑ **clog** gestioneaza iesirea catre dispozitivul standard de eroare (ecranul).

**cin** (“see in”) = standard input stream  
**cout** (“see out”) = standard output stream  
Include in `<iostream>`

**>>** (“put to”) = operator de insertie (cin)  
**<<** (“get from”) = operator de extractie (cout)

# 7.2 Utilizarea cin si cout

## Using namespace std

### SINTAXA

In C++ pentru cin si cout se include biblioteca <iostream> :

```
#include <iostream>  
using namespace std;  
cin >> a;  
cout<< a;  
...
```

Sau in varianta :

```
#include <iostream>  
std::cin >> a;  
std::cout<<a;
```

**Obs:** scrierea <iostream.h> nu este suportata de anumite compilatoare => se va include <iostream>

# 7.2 Utilizarea cin si cout

## Afisarea datelor in C++: **cout**

### SINTAXA

**Format de declarare:**

```
cout << "text"<< var1 << var2 ...<<"\n"<<endl;
```

unde : **var1, var2, ...** pot fi **nume de variabile, constante sau expresii** ;

Trebuie inclusa biblioteca <iostream>

**Efect:**

- Afiseaza pe ecran textul sau valoarea variabilei, constantei sau rezultatul evaluarii expresiei de dupa **operatorul de extractie <<**
- "\n" si endl realizeaza trecerea la un rand nou

# 7.2 Utilizarea cin si cout

## Afisarea datelor in C++: **cout**

**Ex. 1:** operatii de scriere (afisare) echivalente

```
cout << "Hello \n";  
cout << "Hello" << endl;
```

**EXAMPLE**

**Ex.2 :** scrierea (afisarea) unui text , constante, variabile

```
cout << "Programare in C++"; // tipareste pe ecran Programare in C++  
cout << 120; // tipareste valoarea 120 pe ecran  
cout << x; // tipareste valoarea x pe ecran
```

**Ex.3 :** afisarea cu extractie multipla (include text explicativ)

```
cout << i << "la patrat este" << i*i << "\n";
```

**Efect:** daca  $i=10$  se afiseaza **10 la patrat este 100**

# 7.2 Utilizarea cin si cout

## Afisarea datelor in C++: cout

```
#include <iostream>
using namespace std;
int main() {
int x=10, y=25, z;
z=x+y;
cout<<"x="<<x <<" y="<<y<<endl;
cout<<"Sum of x+y = " <<z;
}
```

Ce se afiseaza pe ecran?

x=10 y=25

Sum of x+y = 35

Se pot combina I/O in C++?

```
#include <iostream>
#include <stdio.h>
using namespace std;
int main()
{ printf("hello! ");
  cout << "Hello world!" << endl;
  return 0;
}
```

Se poate utiliza si <cstdio> in loc de <stdio.h>:

```
#include <cstdio>
```

# 7.2 Utilizarea cin si cout

## Citirea datelor in C++: **cin**

### SINTAXA

Format de declarare: **cin >> var1 >> var2 ...;**

unde : **var1, var2, ...** pot fi numai nume de variabile **nu si constante sau expresii**; trebuie inclusa biblioteca <iostream>

**Efect:** daca variabila citita de la tastatura este de tip:

- char:** se citeste un singur caracter diferit de blank (spatiu) !!
- int:** se citeste o constanta intreaga optional precedata de un semn
- float:** se citeste o constanta intreaga sau reala (eventual specificata in format stiintific), optional precedata de semn



# 7.2 Utilizarea cin si cout

## Citirea datelor in C++: cin

### EXEMPLE

**Ex.1** : citirea unui intreg : `int i; cin >> i;` ⇔ `int i; scanf("%d", &i);`

**Ex.2** : urmatoarele instructiuni sunt echivalente

`int i,n; cin >> i >> n;` ⇔ `int i,n; cin >> i;cin >>n;`

**Ex.3** : daca `int i,j; char ch;` urmatoarele instructiuni de citire au urmatorul efect:

Instructiuni	Data intrare	Continut variabile dupa insertie
1. <code>cin &gt;&gt; i;</code>	32	i=32
2. <code>cin &gt;&gt; i &gt;&gt; j;</code>	4 60	i=4, j=60
3. <code>cin &gt;&gt; i &gt;&gt; ch &gt;&gt; x;</code>	25 A 16.9	i=25, ch='A', x=16.9
4. <code>cin &gt;&gt; i &gt;&gt; ch &gt;&gt; x;</code>	25 A 16.9	i=25, ch='A', x=16.9
5. <code>cin &gt;&gt; i &gt;&gt; ch &gt;&gt; x;</code>	25A16.9	i=25, ch='A', x=16.9
6. <code>cin &gt;&gt; i &gt;&gt; j &gt;&gt; x;</code>	12 8	i=12, j=8 –calculatorul asteapta introducerea lui x
7. <code>cin &gt;&gt; i &gt;&gt; x;</code>	46 32.4 15	i=46,x=32.4 –calculatorul retine ultimul nr. (15) pentru o insertie ulterioara; daca aceasta nu se realizeaza se pierde

# 7.2 Utilizarea cin si cout

## Exemple citire/afisare in C++

### EXEMPLE

Ex.4 : citirea unui intreg :

```
#include <iostream>
using namespace std;
int main ()
{ int i;
cout << "Introduceti o valoare intreaga:\n ";
cin >> i;
cout << "Valoarea citita este: " << i;
cout << " \niar dublul ei este: " << i*2 << ".\n";
return 0;}
```

```
Introduceti o valoare intreaga:
10
Valoarea citita este: 10
iar dublul ei este: 20.
```

Ce se intampla daca utilizatorul introduce o valoare care nu este de tip intreg?

R: Eroare la citire si variabila i ramane neinitializata producand rezultate incorecte.

# 7.2 Utilizarea cin si cout

## Exemple citire/afisare cu apel functii in C++

### EXEMPLE

Ex. : calculul patratului si cubului unui nr. intreg utilizand cin si cout

```
#include <iostream>
using namespace std;
int Patrat( int );
int Cub( int );
int main()
{int a;
cout << "a= " ;cin >> a ;
cout << "patratul lui " << a << " este " << Patrat(a) << endl;
cout << "iar cubul lui " << a << " este " << Cub(a) << endl;
return 0;}
int Patrat( int n )
{ return n * n;}
int Cub( int n )
{ return n * n * n; }
```

```
a= 10
patratul lui 10 este 100
iar cubul lui 10 este 1000
```

# 7.2 Utilizarea cin si cout

## Exemple citire/afisare tablouri in C++

### EXEMPLE

Ex. : citirea si afisarea inversa a unui sir de numere reale

```
#include <iostream>
using namespace std;
int main()
{double a[20];
int i, n;
cout<<"Lungimea sirului n="; cin>>n;
//Citire sir de numere reale
for (i=0; i<n; i++)
{   cout<<"a["<<i<<"]=";cin>>a[i]; }
//Afisarea elementelor sirului:
cout<<"Sirul introdus este:\n";
for (i=0; i<n ;i++) cout<<a[i]<<" ";
//Afisarea elementelor sirului în ordine inversa:
cout<<"\nSirul inversat este:\n";
for (i=n-1; i>=0; i--) cout<<a[i]<<" ";
cout <<endl; return 0;}
```

```
Lungimea sirului n=5
a[0]=1
a[1]=5
a[2]=2
a[3]=3
a[4]=4
Sirul introdus este:
1,5,2,3,4,
Sirul inversat este:
4,3,2,5,1,
```

# 7.2 Utilizarea cin si cout

## Exemple citire/afisare tablouri in C++

### EXAMPLE

Ex. : citirea si afisarea elementelor unei matrici de numere reale

```
#include <iostream>
using namespace std;
int main()
{int A[10][10],i,j,n,m;
cout<<"Nr. linii:"; cin>>n;
cout<<"Nr. coloane:"; cin>>m;
//citirea elementelor matricii
for (i=0; i<n; i++)
    for (j=0; j<m; j++)
        { cout<<"A["<<i<<","<<j<<"]="";
          cin>>A[i][j];}
//afisarea elementelor matricii
for (i=0; i<n; i++)
    {for (j=0; j<m; j++)
      cout<<A[i][j]<<"\t";cout<<"\n";}
return 0;}
```

```
Nr. linii:2
Nr. coloane:3
A[0,0]=1
A[0,1]=2
A[0,2]=3
A[1,0]=4
A[1,1]=5
A[1,2]=6
1      2      3
4      5      6
```

# 7.2 Utilizarea cin si cout

## Exemple citire/afisare utilizand pointeri in C++

### EXEMPLE

**Ex. :** Programul citește un sir de nr. intregi de la tastatura, și înlocuiește elementul maxim cu o valoare introdusă de la tastatură, utilizand pointeri.

```
#include <iostream>
using namespace std;
int main()
{ int a[20], *p, n,i, max, indice, val; p=a;
cout<<"Nr. elemente:"; cin>>n;
for (i=0; i<n; i++) { cout<<"a["<<i<<"]=""; cin>>*(p+i);}
// citirea elementelor sirului
max= *p; indice=0;
for (i=0; i<n; i++) if (max<= *(p+i)){ max= *(p+i); indice=i;}
    cout<<"Maxim :"<<max<<" , pozitia in sir:"<<indice+1<<endl;
// determinarea valorii elementului maxim din sir si a pozitiei acestuia
cout<<"Valoare de inlocuire:"; cin >> val;
*(p+indice)=val; //afisarea noului sir
for (i=0; i<n; i++) cout<< *(p+i)<<"\t";cout<<"\n"; return 0;}
```

```
Nr. elemente:5
a[0]=1
a[1]=8
a[2]=10
a[3]=2
a[4]=3
Maxim :10, pozitia in sir:3
Valoare de inlocuire:100
1      8      100      2      3
```

# 7.2 Utilizarea cin si cout

## Exemple citire/afisare structuri in C++

### EXAMPLE

Ex. : Programul citește numele , prenumele si 4 note pentru fiecare student si afiseaza studentii sortati in ordinea mediilor

```
#include <iostream>
using namespace std;
struct student {char nume[20], prenume[20];double nota[4]; double media;} s[100], aux[100];
int i=0,j,n,k;double sum;
int main (void)
{cout <<"Introduceti numar studenti:";cin>>n;
for (i=0;i<n;i++)
    { cout<<"\nIntroduceti numele studentului "<<i+1<<":";cin>>s[i].nume;
      cout<<"Introduceti prenumele studentului "<<i+1<<":"; cin>>s[i].prenume;
      for (sum=0,j=0;j<4;j++)
          {cout<<"Introduceti nota:"<<j+1<<" a studentului: "<<i+1<<":";
            cin>>s[i].nota[j]; sum+=s[i].nota[j];}   s[i].media=sum/4.;}
cout<<"\nLista studentilor:\n";
for (i=0;i<n;i++) cout<<s[i].nume<<"\t"<<s[i].prenume<<"\t"<<s[i].media<<endl;
cout<<"\nStudentii sortati dupa medii:";
do {k=0; for (i=0;i<n-1;i++)
    { if (s[i].media<s[i+1].media){ aux[i]=s[i] ;s[i]=s[i+1];s[i+1]=aux[i];k=1;}} while (k);
for (i=0;i<n;i++) { cout<<s[i].nume<<"\t"<<s[i].prenume<<"\t"<< s[i].media<<endl;
return 0;}
```

```
Introduceti numar studenti:2
Introduceti numele studentului 1:Pop
Introduceti prenumele studentului 1:Alin
Introduceti nota:1 a studentului: 1:5
Introduceti nota:2 a studentului: 1:8
Introduceti nota:3 a studentului: 1:6
Introduceti nota:4 a studentului: 1:7

Introduceti numele studentului 2:Rusu
Introduceti prenumele studentului 2:Radu
Introduceti nota:1 a studentului: 2:10
Introduceti nota:2 a studentului: 2:9
Introduceti nota:3 a studentului: 2:10
Introduceti nota:4 a studentului: 2:8

Lista studentilor:
Pop      Alin      6.5
Rusu     Radu      9.25

Studentii sortati dupa medii:
Rusu     Radu      9.25
Pop      Alin      6.5
```



# TEST

1 . Considerand instructiunile : `double x=9, y[10];y[9]=x-2;`  
Indicati care dintre instructiunile de mai jos are ca efect afisarea pe ecran a urmatoarelor informatii `x=9,y[9]=7` :

- a) `cin>>"x=">>x>>"y[">>x>>"]=">>y ;`
- b) `cout<<x<<y ;`
- c) `cout<<x=9<<"y[9]="<<7;`
- d) `cout<<"x="<<x<<"y["<<9<<"]="<<y[9] ;`

**Raspuns correct**

**d)**





### Raspuns correct

c) Nu se pot utiliza expresii sau constante in cin

## TEST

2 . Care este rezultatul acestui program daca de la tastatura se introduce: 60?

```
#include <iostream>
using namespace std;
int main ()
{int i;
cout << "Introduceti o valoare intreaga: ";
cin >> i + 4;
cout << i; return 0;}
```

- a) 64
- b) i+4
- c) Eroare
- d) Ambele variante a) si b)

# 7.3 Citirea/afisarea cu format in C++

## Citire/afisare cu format in C++

Cum citim si afisam date cu format in C?

Utilizand `scanf` si `printf`

Cum citim si afisam date cu format in C++?

Utilizand functii speciale

`cin` citeste si spatiile sau le ignora?

`cin` ignora spatiile

Cum citim in C++ o variabila `char` daca este " " (spatiu)?

Utilizand `cin.get()`

# 7.3 Citirea/afisarea cu format in C++

Citirea caracterelor si sirurilor de caractere : **cin.get()**

## SINTAXA

Format de apelare a functiei pentru a prelua un caracter :

```
cin.get(caracter);
```

Efect: preluare **caracter** de la tastatura indiferent daca acesta este spatiu sau '\n'

Format de apelare a functiei pentru a prelua un sir de caractere :

```
cin.get(sir_caractere, [int dim_max_sir ], [caracter final]);
```

Efect: preluare **sir\_caractere** (=cuvant pana la primul spatiu sau delimitator= caracter final sir) de la tastatura de dimensiune maxima dim\_max\_sir specificata optional

# 7.3 Citirea/afisarea cu format in C++

## Citirea caracterelor : **cin.get()**

### EXEMPLU

Ex. : Diferenta cin si cin.get(): **cin ignora spatiile si caracterul linie noua**  
**cin >>ch1 >>ch2>>ch3;**

Daca se introduce de la tastatura:

**R 1**

atunci lui ch1 i se atribuie caracterul "R" se sare peste spatiu si se atribuie lui ch2 caracterul "1" , ch3 ramane neinitializat

Daca se doreste si citirea spatiului dintre R si 1 atunci se utilizeaza cin.get()

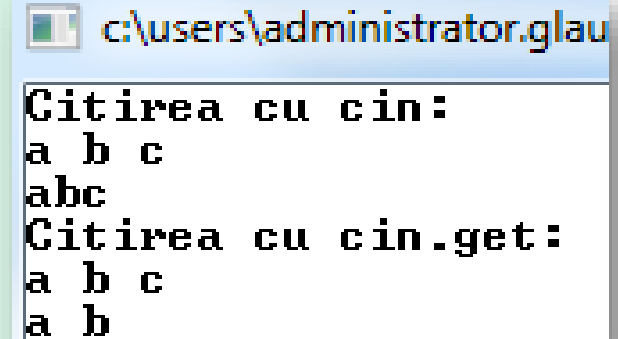
```
cin .get(ch1);  
cin .get(ch2);  
cin .get(ch3);
```

# 7.3 Citirea/afisarea cu format in C++

## Citirea caracterelor : **cin.get()**

### EXEMPLU

```
#include <iostream>
using namespace std;
int main()
{
char ch1,ch2,ch3;
cout<<"Citirea cu cin:"<<endl;
cin >> ch1 >> ch2 >> ch3;
cout<<ch1 << ch2 << ch3;
cout<<"\nCitirea cu cin.get:"<<endl;
cin >> ch1;
cin .get(ch2);
cin >> ch3;
cout<<ch1 << ch2 << ch3; return 0;
}
```



```
c:\users\administrator.gla...
Citirea cu cin:
a b c
abc
Citirea cu cin.get:
a b c
a b
```

## 7.3 Citirea/afisarea cu format in C++

**Citirea sirurilor de caractere :** `cin.get(sir caractere, [int dim_max_sir ], [delimitator]);`

### EXEMPLU

**Ex. :** Preluarea unui sir de la tastatura de maxim 255 caractere, fara delimitator explicit. Implicit “\n” este considerat delimitator.

```
#include <iostream>
using namespace std;
int main()
{
    char sir[255];
    cout << "Introduceti un sir: ";
    cin.get(sir,255);
    cout << "Sirul este:" << sir << endl;
}
```

```
Introduceti un sir
Programare in C/C++
Sirul este:Programare in C/C++
```

## 7.3 Citirea/afisarea cu format in C++

**Citirea sirurilor de caractere :** `cin.get(sir caractere, [int dim_max_sir ], [delimitator]);`

### EXEMPLU

**Ex. :** Preluarea unui cuvant de la tastatura de maxim 255 caractere care se termina cu \* (cu delimitator)

```
#include <iostream>
using namespace std;
int main()
{
char sir[255];
cout << "Introduceti un sir: ";
cin.get(sir,255, '*');
cout << "Sirul este:" << sir << endl;
}
```

```
Introduceti un sir
Programare in C/C++*
Sirul este:Programare in C/C++
```

# 7.3 Citirea/afisarea cu format in C++

Citirea sirurilor de caractere : **cin.getline()**

## SINTAXA

```
void cin.getline(char *s, const int size);
```

Format de apelare a functiei :

```
cin.getline (sir caractere, dimensiune sir)
```

```
cin.getline (sir caractere, dimensiune sir, [delimitator])
```

**Efect:** preluarea unui sir de caractere de la tastatura de dimensiune specificata sau pana la întâlnirea caracterului delimitator si stocarea lui in variabila sir

**Ex. : fara delimitator**

```
#include <iostream>
```

```
using namespace std;
```

```
int main ()
```

```
{ char mybuffer[100];
```

```
cout << "Cum te cheama? "; cin.getline (mybuffer,100);
```

```
cout << "Buna ziua " << mybuffer << ".\n";
```

```
cout << "Care este autorul tau preferat? "; cin.getline (mybuffer,100);
```

```
cout <<"OK. " << mybuffer <<"este un autor clasic" <<"\n"; return 0;}
```

```
Cum te cheama? Popa Anca
```

```
Buna ziua Popa Anca.
```

```
Care este autorul tau preferat? Ernest Hemingway
```

```
OK.Ernest Hemingway este un autor clasic
```

**EXEMPLU**



# 7.3 Citirea/afisarea cu format in C++

## Citirea sirurilor de caractere : **cin.getline()**

### EXEMPLU

**Ex. :cu delimitator** , citirea unui sir de caractere de la tastatura pana se intalneste un caracter "t" sau s-au citit 100 caractere (prima conditie indeplinita este luata in considerare)

```
#include <iostream>
using namespace std;
int main()
{
char line[100];
cout << " Introduceti un text terminat cu 't' " << endl;
cin.getline( line, 100, 't' );
cout << line;
}
```

Ce se afiseaza daca se introduce textul

Introduceti un sir terminat cu 't':  
>Test Programare in C++

Se afiseaza:  
>Tes

# 7.3 Citirea/afisarea cu format in C++

## Citirea sirurilor de caractere : **cin.getline()**

### EXEMPLU

Ex. : Concatenarea a 2 siruri citite de la tastatura utilizand biblioteca <cstring>

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{char c1[20], c2[20];
cout << "Sirul 1 este:" << endl;
cin.getline(c1,20);
cout << "Sirul 2 este:" << endl;
cin.getline(c2,20);
cout << "Sirurile concatenate:" << endl;
cout << strcat(c1,c2);
return 0;}
```

```
Sirul 1 este:
Programare anul I
Sirul 2 este:
 C si C++
Sirurile concatenate:
Programare anul I C si C++
```

# 7.3 Citirea/afisarea cu format in C++

Ignorarea unor caractere citite : **cin.ignore()**

## SINTAXA

Format de apelare a functiei : **cin.ignore(intreg, caracter);**

Unde:

- intreg** reprezinta un nr. intreg sau o expresie de tip intreg
- caracter** este o constanta de tip caracter

**Efect:** citirea cu ignorarea unui nr. de caractere introduse de la tastatura sau pana la intalnirea caracterului specificat.

**Prima conditie indeplinita se ia in considerare!**

# 7.3 Citirea/afisarea cu format in C++

## Afisarea datelor cu cout in diverse baze de numeratie

### SINTAXA

Afisarea unui nr. in baza 10,16,8 se realizeaza utilizand modifierii: **dec, hex , oct**

- dec** pentru reprezentarea in baza 10
- hex** pentru reprezentarea in baza 16
- oct** pentru reprezentarea in baza 8

Modifierii dec, hex si oct se utilizeaza dupa cout si inaintea variabilei ce se va tipari in baza de numeratie respective. **Efect: conversia valorilor specificate** dupa modifieri in baza de numeratie

**Ex. :** citeste un nr intreg de la tastatura si se afiseaza in baza 10, 8 si 16

```
#include <iostream>
using namespace std;
void main(void)
{ int nr; cout<<"nr=";cin >> nr;
cout << "valoare zecimala:" << dec <<nr<<endl;
cout << "\nvaloare octala:" << oct << nr <<endl;
cout << "\nvaloare hexazecimala:"<< hex << nr;}

```

```
nr=54
valoare zecimala:54
valoare octala:66
valoare hexazecimala:36

```

### EXEMPLE

# 7.3 Citirea/afisarea cu format in C++

Afisarea datelor cu format (nr minim de caractere afisate): **cout.width**

**SINTAXA**

Format : **cout.width(nrmin)**

❑ **nrmin** = nr. minim de caractere utilizate pentru afisare

**Efect:** datele de iesire se afiseaza pe **nrmin** de caractere

**Obs:** **width** trebuie specificat pentru fiecare valoare de iesire. `cout.width()` are effect doar pentru urmatorul `cout` nu si pentru celelalte `cout` din program

**Ex. :** afisarea nr. de la 1 la 3 pe cate 5 pozitii fiecare

```
#include <iostream>
using namespace std;
int main(void)
{ int i;
  for (i=0;i<3;i++)
    {cout.width(5);
     cout <<i<<'\n';} return 0;}
```

```
0
1
2
```

**EXEMPLE**

# 7.3 Citirea/afisarea cu format in C++

**Afisarea datelor cu format (nr minim de caractere afisate): `setw`**

## SINTAXA

**Format :** `setw(nrmin)`

☐ `nrmin` = nr. minim de caractere utilizate pentru afisare

**Efect:** datele de iesire sunt afisate pe `nrmin` de caractere, inclusa in `<iomanip>`

## EXEMPLE

**Ex. :** afisarea nr. de la 1 la 4 fiecare cu nr diferit de pozitii de afisare

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{ int i;
  for (i=1;i<5;i++)
  cout << "\n"<< setw(i) <<i;
  return 0;}
```

```
1
 2
   3
    4
```

# 7.3 Citirea/afisarea cu format in C++

Afisarea datelor cu format (caractere de umplere): **cout.fill**

SINTAXA

Format : **cout.fill(caracter)**

❑ **caracter** = caracterul utilizat pentru completarea datelor de iesire

Efect: se utilizeaza impreuna cu **cout.width()** , iar datele de iesire sunt afisate pe nrmin de caractere, caracterele necomplete se inlocuiesc cu **caracter**

EXEMPLE

Ex. : afisarea nr. de la 0 la 2 cu umplerea spatiilor ramase libere cu puncte

```
#include <iostream>
using namespace std;
int main(void)
{ int i;
  for (i=0;i<3;i++)
    {cout.fill('.');
     cout.width(5+i);
     cout <<i<<'\n';}return 0;}
```

```
.....0
.....1
.....2
```

# 7.3 Citirea/afisarea cu format in C++

**Afisarea datelor cu format (cu zecimale pentru nr reale): `setprecision()`**

## SINTAXA

**Format :** `setprecision(nr)`

❑ `nr` = de cifre utilizate pentru afisarea datelor de tip float sau double

**Efect:** datele de iesire sunt afisate cu nr de cifre semnificative indicat de `nr`

Acest specificator de format este inclus in `<iomanip.h>`. Se utilizeaza inaintea variabilei ce trebuie tiparite

## EXEMPLE

**Ex. :** afisarea nr. 7.3145 cu 1,2,3 zecimale

```
#include <iostream>
#include <iomanip>
using namespace std;
int main(void)
{ int i;float val=7.3145;
  for(i=1;i<5;i++)
  cout << setprecision(i) <<val <<'\n';
  return 0;}
```

```
7
7.3
7.31
7.314
```



# 7.3 Citirea/afisarea cu format in C++

## Afisarea datelor cu format fix sau stiintific: **fixed**, **scientific**

### SINTAXA

Format : **std::fixed** **std::scientific**

- ❑ **fixed** = indica afisarea datelor reale cu numarul de zecimale implicit (6)
- ❑ **scientific** = indica afisarea datelor reale in format stiintific

**Efect:** datele de iesire sunt afisate in format clasic cu zecimale (**fixed**) sau in format stiintific (**scientific**)

### EXEMPLE

Ex. : afisarea in format fix sau stiintific

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{ double a = 3.1415926534; double b = 1.0e-10;
  cout <<fixed<< a << '\n';
  cout <<scientific << b << '\n';
  return 0;}
```

```
3.141593
1.000000e-10
```

# 7.3 Citirea/afisarea cu format in C++

## Afisarea datelor cu aliniere stanga/dreapta: **left/right**

### SINTAXA

Format : **std::left** **std::right**

- ❑ **left** = indica afisarea datelor cu aliniere la stanga
- ❑ **right** = indica afisarea datelor cu aliniere la dreapta

**Efect:** datele de iesire sunt afisate cu aliniere la stanga/dreapta

### EXEMPLE

Ex. : afisarea cu aliniere la stanga/dreapta

```
#include <iostream>
#include <iomanip>
using namespace std;
int main ()
{int i;
cout << "Aliniere dreapta\n";
for (i=10;i<15;i++) {cout.width(4); cout<<right <<i;}
cout << "\nAliniere stanga\n";
for (i=10;i<15;i++) {cout.width(3); cout << left <<i;}
return 0;}
```

```
Aliniere dreapta
 10 11 12 13 14
Aliniere stanga
10 11 12 13 14
```



# TEST kahoot

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

<http://kahoot.it>