

C++

Introducere în Programare

Un ghid prietenos pentru clasele V-VIII

 Comentarii & Consolă

 Variabile & Tipuri

 Operatori

 Condiționale (if)

 Repetitive (for, while)

 Exerciții rezolvate



Cuprins

Găsești mai jos lista completă a capitolelor. Fiecare capitol are explicații, exemple și exerciții.





1	Ce este C++? Primul contact	3
2	Primul program & comentarii	4
3	Consola: cout și cin	6
4	Variabile și tipuri de date	8
5	Operatori	10
6	Instrucțiuni liniare	12
7	Instrucțiuni condiționale (if)	13
8	Instrucțiuni repetitive (for, while)	15
9	Recapitulare & provocări finale	18

1 Ce este C++?

Imaginează-ți că vrei să-i spui calculatorului să deseneze, să calculeze sau să joace un joc. Nu-l poți ruga în română — trebuie să-i vorbești pe limba lui. Exact asta e un **limbaj de programare**: un mod de a-i da instrucțiuni calculatorului.

De ce C++?

C++ este unul dintre cele mai populare limbaje de programare din lume. Cu el sunt construite:

-  Jocuri video (Minecraft, Counter-Strike, majoritatea jocurilor AAA)
-  Browsere web (Google Chrome, Mozilla Firefox)
-  Programele din mașini, avioane, roboți
-  Olimpiadele de informatică — este limbajul oficial!

Știi că...?

C++ a fost creat în 1979 de **Bjarne Stroustrup**. Numele vine de la operatorul `++` care înseamnă „adună unu”. Deci C++ înseamnă „un C mai bun”.

Cum „gândește” un program?

Un program rulează pas cu pas, de sus în jos, ca o rețetă de bucătărie. Dacă îi ceri:

1. Ia un ou.
2. Sparge-l în tigaie.
3. Pune sare.

Calculatorul execută **exact** în ordinea asta. Dacă uiți un pas — sau îl scrii greșit — nu-ți iese omleta. La fel și cu programul: o literă greșită și nu mai merge!

Sfat de la prof

Nu-ți fie teamă de erori. **Erorile sunt prietenele tale** — ele îți spun unde ai greșit. Fiecare programator, chiar și cel mai bun din lume, greșește zilnic.

2 Primul program & comentarii

👋 „Salut, lume!”

Prin tradiție, primul program pe care îl scrie orice programator din lume afișează pe ecran mesajul „**Salut, lume!**”. Hai să-l scriem și noi:

```
// Primul meu program C++
#include <iostream>
using namespace std;

int main() {
    cout << "Salut, lume!";
    return 0;
}
```

Consolă

```
Salut, lume!
```

🔍 Să disecăm fiecare linie

Linia	Ce face
<code>#include <iostream></code>	„Adu-mi unelta pentru scris/citit pe ecran”
<code>using namespace std;</code>	„Nu mă obliga să scriu <code>std::</code> la fiecare comandă”
<code>int main()</code>	Funcția principală — aici începe programul
<code>{ ... }</code>	Acoladele marchează începutul și sfârșitul programului
<code>cout << "..."</code>	„Scrie pe ecran textul dintre ghilimele”
<code>return 0;</code>	„Gata, programul s-a terminat cu succes”

⚠️ Atenție!

Fiecare instrucțiune se termină cu `;` (punct și virgulă). Dacă uiți, programul nu compilează! E ca punctul la sfârșit de propoziție — fără el, calculatorul se încurcă.

💬 Comentariile — notițe pentru tine

Comentariile sunt texte pe care le scrii în program **doar pentru tine sau pentru cine citește codul**. Calculatorul le ignoră complet.

```
// Acesta este un comentariu pe o singură linie

/* Iar acesta este un comentariu
   pe mai multe linii.
   Util pentru explicații lungi. */
```

```
int varsta = 13; // poți scrie și la sfârșit de linie
```

✓ De ce sunt utile?

- Îți explici ție însuși ce face codul (peste o săptămână ai uitat!)
- Poți „dezactiva” temporar o linie punând `//` în față
- Ajuți colegii care îți citesc programul

✍ Exercițiu rezolvat 2.1

Enunț: Scrie un program care afișează pe ecran numele și clasa ta, pe două rânduri diferite.

✓ Rezolvare

```
#include <iostream>
using namespace std;

int main() {
    cout << "Ma numesc Andrei" << endl;
    cout << "Sunt in clasa a VII-a";
    return 0;
}
```

`endl` înseamnă „trece pe rândul următor” (end of line).

? Exerciții propuse

1. Scrie un program care afișează „Astăzi învăț C++!” pe ecran.
2. Afișează pe trei rânduri: numele școlii, clasa, numele profesorului.
3. Adaugă în programul de mai sus un comentariu care explică ce face fiecare linie.

3 Consola: cout și cin

Consola este „fereastra neagră” prin care programul tău comunică cu tine. Cu `cout` *scrii* pe ecran, cu `cin` *citești* ce tastează utilizatorul.

cout — scriere pe ecran

Cuvântul vine de la Character *OUT*put („ieșire de caractere”). Folosește săgețile `<<` îndreptate **spre cout** (ca și cum îi dai ceva).

```
cout << "Am" << 13 << " ani" << endl;
cout << "Clasa a VII-a B";
```



```
Am 13 ani
Clasa a VII-a B
```

Trucul săgeților

Gândește-te la `<<` ca la o pâlnie: informația „curge” de la dreapta spre stânga, direct în `cout`. Poți pune oricâte „curgeri” vrei pe o linie.

cin — citire de la tastatură

Cuvântul vine de la Character *IN*put. Săgețile sunt invers: `>>` — informația vine **din cin către variabila ta**.

```
int varsta;
cout << "Cati ani ai? ";
cin >> varsta;
cout << "Ai " << varsta << " ani!";
```



```
Cati ani ai? 13
Ai 13 ani!
```

Formatare simplă

Simbol	Ce face	Exemplu
<code>endl</code>	Trece la linie nouă	<code>cout << "a" << endl << "b";</code>
<code>"\n"</code>	Linie nouă (mai scurt)	<code>cout << "a\nb";</code>
<code>"\t"</code>	Un TAB (spațiu mare)	<code>cout << "a\tb";</code>

Exercițiu rezolvat 3.1

Enunț: Citește de la tastatură numele și vârsta utilizatorului, apoi afișează un mesaj de salut personalizat.

Rezolvare

```
#include <iostream>
using namespace std;

int main() {
    string nume;
    int varsta;

    cout << "Cum te numesti? ";
    cin >> nume;
    cout << "Cati ani ai? ";
    cin >> varsta;

    cout << "Salut, " << nume
         << "! La anul vei avea "
         << varsta + 1 << " ani.";
    return 0;
}
```

Exerciții propuse

1. Citește 2 numere și afișează-le pe două rânduri diferite.
2. Cere utilizatorului orașul în care locuiește și afișează „Salutări din [oraș]!”.
3. Afișează pe ecran — folosind `\t` — un mic tabel: nume, prenume, vârstă (aliniată pe coloane).

4

Variabile și tipuri de date

O **variabilă** este o cutie în care ții o informație. Fiecare cutie are un nume (ca să știi care e a ta) și o etichetă care spune ce fel de lucruri intră în ea (numere întregi? numere cu virgulă? text?). Eticheta se numește **tip de dată**.

Cum declari o variabilă?

```
int varsta = 13;           // cutie pentru numere intregi
double nota = 9.75;      // cutie pentru numere cu virgula
char initiala = 'A';     // cutie pentru o singura litera
bool elevBun = true;     // cutie pentru adevarat/fals
string nume = "Maria";   // cutie pentru text (necesita #include <string>)
```

Tipurile principale

Tipul	Ce ține	Exemple de valori	Memorie
<code>int</code>	numere întregi	-5, 0, 42, 1000	4 octeți
<code>float</code>	numere cu virgulă (precizie mică)	3.14, -0.5	4 octeți
<code>double</code>	numere cu virgulă (precizie mare) ★	3.14159265, 9.81	8 octeți
<code>char</code>	un singur caracter	'A', '7', '!	1 octet
<code>bool</code>	adevărat sau fals	true, false	1 octet
<code>string</code>	șir de caractere (text)	"Salut", "C++"	variabil

✓ Când folosești ce?

- Ai de-a face cu **ceva care se numără** (vârstă, nr. elevi, puncte)? → `int`
- Ai de-a face cu **ceva care se măsoară** (greutate, temperatură, notă)? → `double`
- Ai de-a face cu **text**? → `string`
- Ai de-a face cu **o alegere da/nu**? → `bool`

👉 Reguli pentru nume de variabile

✓ Permis

- Litere (a-z, A-Z)
- Cifre (0-9)
- Caracterul `_`
- Trebuie să înceapă cu literă

✗ Nepermis

- Spații: `nota elev`
- Să înceapă cu cifră: `2nota`
- Diacritice: `vârstă`
- Cuvinte rezervate: `int`, `if`

! C++ face diferență între majuscule și minuscule!

Nota , nota și NOTA sunt trei variabile complet diferite!

Exercițiu rezolvat 4.1

Enunț: Citește trei note ale unui elev și afișează media aritmetică cu 2 zecimale.

Rezolvare

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    double n1, n2, n3;
    cout << "Introdu 3 note: ";
    cin >> n1 >> n2 >> n3;

    double media = (n1 + n2 + n3) / 3;
    cout << fixed << setprecision(2);
    cout << "Media este: " << media;
    return 0;
}
```

`setprecision(2)` afișează doar 2 zecimale.

Exerciții propuse

1. Citește lungimea și lățimea unui dreptunghi. Afișează aria și perimetrul.
2. Citește o sumă în lei și afișează cât reprezintă în euro (1 EUR = 5 RON).
3. Citește o temperatură în Celsius și afișează-o în Fahrenheit.

Formula: $F = C \times 9/5 + 32$

5 Operatori

Operatorii sunt „uneltele” cu care calculezi, compari sau combini valori. Există mai multe categorii — hai să le vedem pe toate!

+ Operatori aritmetici

Op.	Nume	Exemplu	Rezultat
+	adunare	<code>7 + 3</code>	10
-	scădere	<code>7 - 3</code>	4
*	înmulțire	<code>7 * 3</code>	21
/	împărțire	<code>7 / 3</code>	2 (la int!) / 2.33 (la double)
%	restul împărțirii (modulo)	<code>7 % 3</code>	1

! Capcana împărțirii la întregi

Dacă împarți **două numere întregi**, rezultatul e tot întreg — zecimalele se pierd! `7 / 2` dă **3**, nu 3.5. Ca să obții 3.5, măcar unul dintre numere trebuie să fie **double** : scrie `7.0 / 2` .

1 2 3 4 Operatori de atribuire

```
int x = 10;
x += 5; // echivalent cu x = x + 5; → x devine 15
x -= 3; // x = x - 3; → x devine 12
x *= 2; // x = x * 2; → x devine 24
x /= 4; // x = x / 4; → x devine 6
x++; // x = x + 1; → x devine 7 (incrementare)
x--; // x = x - 1; → x devine 6 (decrementare)
```

⚖ Operatori de comparație (relaționali)

Rezultatul este întotdeauna **true** sau **false** .

Op.	Înseamnă	Exemplu	Rezultat
<code>==</code>	este egal cu	<code>5 == 5</code>	true
<code>!=</code>	diferit de	<code>5 != 3</code>	true
<code><</code>	mai mic decât	<code>3 < 5</code>	true
<code>></code>	mai mare decât	<code>3 > 5</code>	false
<code><=</code>	mai mic sau egal	<code>5 <= 5</code>	true
<code>>=</code>	mai mare sau egal	<code>3 >= 5</code>	false

! Cea mai frecventă greșeală!

`=` înseamnă „atribuie” (pune valoarea). `==` înseamnă „compară”. Nu le încurca: `if (x = 5)` e greșit, `if (x == 5)` e corect!

Operatori logici

Combină mai multe condiții.

Op.	Nume	Când e true
<code>&&</code>	ȘI (and)	Ambele condiții sunt adevărate
<code> </code>	SAU (or)	Cel puțin una e adevărată
<code>!</code>	NU (not)	Inversează: true → false, false → true

```
int varsta = 14;
bool eAdolescent = (varsta >= 13 && varsta <= 19); // true
bool eWeekend = (zi == "sambata" || zi == "duminica");
```

Exercițiu rezolvat 5.1

Enunț: Citește un număr de 2 cifre și afișează suma cifrelor lui.

Rezolvare

```
int n;
cin >> n;
int zeci = n / 10; // ex: 47 / 10 = 4
int unitati = n % 10; // ex: 47 % 10 = 7
cout << zeci + unitati; // 4 + 7 = 11
```

Trucul clasic: `/10` scoate cifra zecilor, `%10` scoate cifra unităților.

? Exerciții propuse

1. Citește un număr de 3 cifre și afișează suma cifrelor.
2. Citește două numere. Afișează `true` dacă sunt ambele pare.
3. Citește vârsta unei persoane. Afișează `true` dacă e copil (sub 12) **sau** senior (peste 65).

6 Instrucțiuni liniare

O instrucțiune liniară este una care se execută **o singură dată**, în ordinea în care e scrisă. Programele simple sunt doar o succesiune de astfel de instrucțiuni.

Structura tipică: Citire → Prelucrare → Afișare



Exercițiu rezolvat 6.1 — Aria cercului

Enunț: Citește raza unui cerc. Afișează aria și circumferința lui.

Rezolvare

```
#include <iostream>
using namespace std;

int main() {
    const double PI = 3.14159;
    double raza, arie, circumf;

    cout << "Raza = ";
    cin >> raza; // 1. CITIRE

    arie = PI * raza * raza; // 2. PRELUCRARE
    circumf = 2 * PI * raza;

    cout << "Aria = " << arie << endl; // 3. AFISARE
    cout << "Circumferinta = " << circumf;
    return 0;
}
```

? Exerciții propuse

1. Citește 2 numere și afișează suma, diferența, produsul și câtul lor.
2. Citește numărul de secunde. Afișează câte ore, minute și secunde reprezintă.
Hint: folosește / și %.
3. Citește prețul unui produs și procentul de reducere. Afișează prețul final.

7 Instrucțiuni condiționale

Până acum, programele noastre mergeau „drept înainte”. Dar în viață luăm decizii: dacă plouă, iau umbrela. Așa și în C++ — folosim instrucțiunea `if`.

Instrucțiunea if

```
if (conditie) {  
    // cod executat DOAR daca conditia e adevarata  
}
```



```
int nota;  
cin >> nota;  
if (nota >= 5) {  
    cout << "Promovat!";  
}
```

if - else

Dacă condiția e falsă, execută **alt** cod.

```
if (nota >= 5) {  
    cout << "Promovat!";  
} else {  
    cout << "Corigent.";  
}
```

if - else if - else (decizii multiple)

```
if (nota >= 9) {  
    cout << "Foarte bine!";  
} else if (nota >= 7) {  
    cout << "Bine.";  
} else if (nota >= 5) {
```

```
    cout << "Suficient.";
} else {
    cout << "Insuficient.";
}
```

✓ Cum gândește C++-ul?

Verifică condițiile **de sus în jos**. De cum găsește una adevărată, execută blocul ei și sare peste toate celelalte. Deci ordinea contează!

📝 Exercițiu rezolvat 7.1 — Paritate

Enunț: Citește un număr și afișează dacă este par sau impar.

✓ Rezolvare

```
int n;
cin >> n;
if (n % 2 == 0) {
    cout << n << " este PAR";
} else {
    cout << n << " este IMPAR";
}
```

Truc clasic: un număr e par dacă restul împărțirii la 2 e zero.

📝 Exercițiu rezolvat 7.2 — Cel mai mare din 3

Enunț: Citește 3 numere. Afișează cel mai mare dintre ele.

✓ Rezolvare

```
int a, b, c;
cin >> a >> b >> c;

if (a >= b && a >= c) {
    cout << a;
} else if (b >= a && b >= c) {
    cout << b;
} else {
    cout << c;
}
```

? Exerciții propuse

1. Citește un număr și afișează dacă e **pozitiv**, **negativ** sau **zero**.
2. Citește vârsta unui elev și afișează clasa probabilă (6 ani → pregătitoare, 7 → cl. I, ..., 14 → cl. VIII).
3. Citește 3 lungimi și verifică dacă pot forma un triunghi.
Condiție: suma oricăror două laturi > a treia.
4. Citește o zi (1-7) și afișează dacă e zi lucrătoare sau weekend.

8 Instrucțiuni repetitive

Ce faci dacă vrei să afișezi numerele de la 1 la 100? Scrii 100 de linii `cout`? Nu! Folosești o **instrucțiune repetitivă** — zisă și **ciclu sau buclă**.

Instrucțiunea for

Folosită când **știi de câte ori** trebuie să se repete.

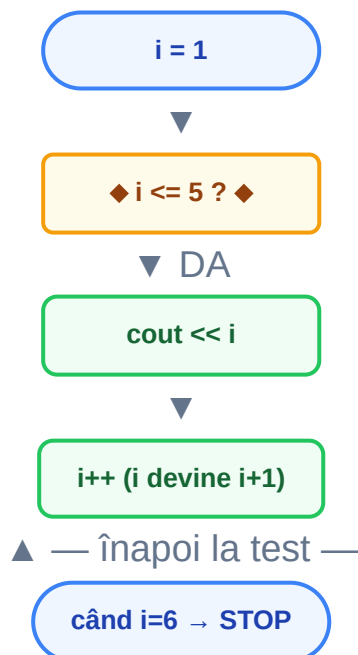
```
for (int i = 1; i <= 5; i++) {  
    cout << i << " ";  
}
```

Consolă

```
1 2 3 4 5
```

Structura for-ului

Partea	Ce face	Când se execută
<code>int i = 1</code>	Inițializare	O singură dată, la început
<code>i <= 5</code>	Condiția de continuare	Înainte de fiecare iterație
<code>i++</code>	Pasul (incrementul)	După fiecare iterație



Instrucțiunea while

Folosită când **nu știi exact de câte ori** se repetă — doar condiția de oprire.

```
int n;  
cout << "Introdu un numar pozitiv: ";  
cin >> n;
```

```
while (n <= 0) {
    cout << "Gresit! Mai incearca: ";
    cin >> n;
}
cout << "Multumesc!";
```

! Atenție la bucla infinită!

Dacă condiția din `while` nu devine niciodată falsă, programul rulează la infinit. Asigură-te că **ceva se schimbă** în interiorul buclei, ca să poată ieși din ea.

do - while

Ca while, dar codul se execută **cel puțin o dată** (testul e la sfârșit).

```
int n;
do {
    cout << "Introdu un nr pozitiv: ";
    cin >> n;
} while (n <= 0);
```

Când folosesc ce?

for

Știi exact de câte ori.
„Afișează primele 10 numere pare.”

while

Repet cât timp se îndeplinește o condiție.
„Cere parolă până o ghicește.”

Exercițiu rezolvat 8.1 — Paritate cu for

Enunț: Citește un număr N și afișează toate numerele **pare** de la 1 la N.

Rezolvare (două metode)

```
// Metoda 1: verific paritatea cu if
for (int i = 1; i <= n; i++) {
    if (i % 2 == 0) {
        cout << i << " ";
    }
}

// Metoda 2: pornesc de la 2 si sar din 2 in 2 (mai eficient!)
for (int i = 2; i <= n; i += 2) {
    cout << i << " ";
}
```

Pentru N = 10, ambele afișează: `2 4 6 8 10`

Exercițiu rezolvat 8.2 — Suma cifrelor

Enunț: Citește un număr natural și afișează suma tuturor cifrelor lui (indiferent câte cifre are).

✓ Rezolvare

```
int n, suma = 0;
cin >> n;

while (n > 0) {
    suma += n % 10; // iau ultima cifra si o adaug
    n = n / 10;    // sterg ultima cifra
}

cout << "Suma cifrelor = " << suma;
```

Exemplu: $n = 3741 \rightarrow 3741 \rightarrow 374 \rightarrow 37 \rightarrow 3 \rightarrow 0$. Suma = $1+4+7+3 = 15$.

? Exerciții propuse

1. Afișează numerele **impare** de la 1 la 100.
2. Citește N. Afișează suma $1+2+3+\dots+N$.
3. Citește N. Afișează N! (factorial: $1 \cdot 2 \cdot 3 \cdot \dots \cdot N$).
4. Citește un număr și numără câte cifre are.
5. Citește un număr și verifică dacă este palindrom (citit invers e la fel: 121, 1331, 474).
6. Afișează tabla înmulțirii cu 7 (de la 7×1 la 7×10).



Harta cunoștințelor

Știu să...	Folosind
Scriu pe ecran	<code>cout <<</code>
Citesc de la tastatură	<code>cin >></code>
Păstrez o valoare	Variabile: <code>int, double, string...</code>
Calculez	Operatori <code>+ - * / %</code>
Iau decizii	<code>if / else if / else</code>
Repet cu număr cunoscut de pași	<code>for</code>
Repet cât timp o condiție e adevărată	<code>while / do-while</code>



Provocări finale



Provocarea 1 — ★ Ușoară

Anul tău de naștere. Citește vârsta utilizatorului și afișează anul probabil al nașterii (consideră anul curent 2026).



Provocarea 2 — ★★ Medie

Divizori. Citește un număr N și afișează toți divizorii lui.

Exemplu: N = 12 → 1, 2, 3, 4, 6, 12.



Provocarea 3 — ★★ Medie

FizzBuzz. Afișează numerele de la 1 la 30. Dacă un nr. e divizibil cu 3 → afișează „Fizz”. Cu 5 → „Buzz”. Cu ambele → „FizzBuzz”. Altfel → numărul.



Provocarea 4 — ★★★ Grea

Număr prim. Citește un număr N și verifică dacă este prim (adică are exact 2 divizori: 1 și el însuși).

Hint: Încearcă să-l împarți la toate numerele de la 2 la N/2.



Provocarea 5 — ★★★★★ Grea

Ghici numărul. Calculatorul „se gândește” la un număr (să zicem 42). Utilizatorul încearcă să-l ghicească. Pentru fiecare încercare, afișezi „mai mare”, „mai mic” sau „ai ghicit!”. Numeri câte încercări a făcut.

💡 **Sfatul final:** Programarea nu se învață citind, se învață **scriind**. Încearcă fiecare exercițiu pe calculator. Greșește. Încearcă din nou. După 10-20 de programe mici, totul va începe să pară natural. Mult succes!



Ai făcut primii pași în C++!

Continuă cu: șiruri de caractere, tablouri, funcții proprii, structuri.

Cele mai bune programe sunt scrise de oameni care continuă să învețe.