

# Elemente de baza ale limbajului C++

---

## 1. Citirea valorilor variabilelor de la tastatura

- Citire variabile de la tastatura:

### 1. Variabila cu variabila:

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

### 2. Citirea tuturor variabilelor intr-un singur apel:

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

- De retinut: valorile se citesc in ordinea in care sunt specificate

- Afisarea valorilor:

- Indiferent daca vorbim de o expresie sau de o variabila, afisarea va functiona in acelasi fel.
- Afisare variabile:

```
int a, b, c;  
// ... presupunem ca variabilele au deja valori  
cout << a << b << c;
```

- De asemenea, la fel precum function `cin`, valorile vor fi afisate in ordinea in care sunt specificate in lista.

- Afisare expresii:

```
cout << 4 * 5 + 34
```

- De retinut:

- Se pastreaza precedenta operatorilor exact precum in exercitiile de matematica
- Daca dorim sa impunem o anumita ordine de efectuare a operatiilor, trebuie sa folosim paranteze. De exemplu, pentru a efectua o adunare inainte de o inmultire:

```
cout << (2 + 5) * 4;
```

## 2. Instructiuni conditionale

- Determina cursul de executie al programului prin evaluarea anumitor conditii.
- In C++ sunt 2 tipuri de instructiuni conditionale:

1. `if / if-else`
2. `switch`

- Structura `if`:

```
if (expresie)
    ruleaza-set-instructiuni
else if (alta expresie)
    ruleaza-alt-set-instructiuni
// aici putem avea oricat de multe conditii.
else
    ruleaza-set-instructiuni-default
// Aceste instructiuni se ruleaza in cazul niciuna dintre conditiile
de mai sus nu au fost evaluate ca fiind adevarate
```

- De retinut:

- dintr-o expresie folosita intr-o instructiune `if` trebuie sa rezulte fie valoarea `true` fie valoarea `false`
- Valoarea `0` este considerata `false`, orice alta valoare este considerata `true`.
  - !!! Valorile negative sunt considerate de asemenea `true`:

```
if (-1) {
    cout << "Acest mesaj va fi afisat";
}
else {
    cout << "Acest mesaj NU va fi afisat.";
}
```

- Desi nu este obligatoriu, chiar daca avem o singura instructiune pentru oricare dintre ramuri (`if`, `else if`, `else`), este indicat sa folosim acoladele!
  - Ramura de `else` sau `else if` trebuie sa fie precedata de o ramura `if` si intotdeauna sa fie ultima ramura.
- Structura `switch`
    - Atunci cand ne trezim in situatia de a avea o succesiune de mai multe `if/else if` instructiuni, unde verificam faptul ca expresia poate avea o valoare anume dintr-un set bine stabilit de valori, este recomandat sa folosim structura `switch`
    - Exemplu: Avem un program care verifica daca valoarea unei variabile este una dintre cele 5 vocale, atunci executam o oarecare instructiune:

```
char ch;
cout << "Enter a character: ";
cin >> ch;
if (ch == 'a') {
    cout << "Ati introdus valoarea a";
}
else if (ch == 'e') {
    cout << "Ati introdus valoarea e";
}
else if (ch == 'i') {
    cout << "Ati introdus valoarea i";
}
else if (ch == 'o') {
    cout << "Ati introdus valoarea o";
}
else if (ch == 'u') {
    cout << "Ati introdus valoarea u";
}
else {
    cout << "Ati introdus un caracter care nu face parte din
[a,e,i,o,u]";
}
```

- Aceasta poate fi rescrisa, folosind structura `switch` precum in bucata de cod de mai jos:

```
char ch;
cout << "Introduceti un caracter: ";
cin >> ch;
switch (ch) {
case 'a':
    cout << "Ati introdus valoarea a";
    break;
case 'e':
    cout << "Ati introdus valoarea e";
    break;
case 'i':
    cout << "Ati introdus valoarea i";
    break;
case 'o':
    cout << "Ati introdus valoarea o";
    break;
case 'u':
    cout << "Ati introdus valoarea u";
    break;
default:
    cout << "Ati introdus un caracter care nu face parte din
[a,e,i,o,u]";
}
```

- Ținând cont de ce am observat în bucata de cod de mai sus, putem deduce următoarea sintaxă pentru structura `switch`:

```
switch (expresie){
    case valoare_1:
        executa_set_de_instructiuni_1;
        break;
    case valoare_2:
        executa_set_de_instructiuni_2;
        break;
    ...
    case valoare_n:
        executa_set_de_instructiuni_n;
        break;
    default:
        executa_set_instructiuni_default;
}
```

## Exercitii propuse

- Pentru o mai bună înțelegere a codului de mai jos, este recomandat să îl evaluați fie prin simplă citire, fie pe o foaie. Comparați rezultatul obținut cu valoarea returnată de către un IDE (de exemplu Code::blocks)
- În cazul în care credeți că instrucțiunile nu sunt valide din punct de vedere C++, gândiți-vă care ar putea fi problema și verificați cu un IDE. De asemenea este recomandat să observați mesajul de eroare returnat de către IDE, pentru a vă dezvolta abilitățile de programare și a rezolva mai ușor problemele similare din viitor.

1. Care va fi rezultatul rularii următoarei bucate de cod:

```
int varsta;
cout << "Introduceti varsta dumneavoastra: ";
cin >> varsta;
if (varsta >= 18) {
    cout << "Sunteti eligibil pentru a incepe scoala de soferi.";
}
else {
    cout << "Va rugam sa mai asteptati...";
}
```

2. Ce se va afișa în urma rularii următoarei bucate de cod:

```
int varsta;
cout << "Introduceti varsta dumneavoastra: ";
cin >> varsta;
if (varsta >= 18) {
    cout << "Sunteti eligibil pentru a incepe scoala de soferi.";
```

```
}  
else {  
    cout << "Va rugam sa mai asteptati...";  
}  
else if (varsta > 9 && varsta < 18) {  
    cout << "Inca sunteti micut pentru a aplica."  
}
```