



# 5. Variabile locale și variabile globale

## 5.1 . Funcția main

În C++ funcția **main** determină prima instrucțiune pe care o va executa programul. Aceasta este unica diferență dintre main și celelalte funcții. Din acest motiv se poate spune că “orice se poate face în main se poate face și în celelalte funcții”.



## 5.2. Variabile locale

La fel cum se declară variabilele în cadrul funcției main, la fel se pot declara variabile și în cadrul celorlalte funcții. Aceste variabile se numesc **locale** și sunt accesibile doar în funcția în care au fost declarate.

În cadrul unei funcții se pot apela și alte funcții, cu condiția ca acestea să fi fost definite înaintea eventualului apel sau este prezent un prototip de funcție înaintea funcției apelate.

## 5.3. Variabile globale

**Variabilele globale** sunt declarate în afara oricărei funcții și sunt vizibile (pot fi utilizate) în tot programul (în programul principal și în subprograme) din momentul declarării lor.

## Lecția 4

molnar1956@yahoo.com



| Exemplul 5.1                                                                                                                                                                                                                       | Exemplul 5.2                                                                                                                                                                                                                                                                           | Exemplul 5.3                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre># include &lt;iostream&gt; using namespace std; int N; void f1() {     int x=5;     N=10;     cout &lt;&lt; endl&lt;&lt;N;     cout &lt;&lt; endl &lt;&lt; x; } int main () {     N=4;     cout &lt;&lt; N;     f1(); }</pre> | <pre># include &lt;iostream&gt; using namespace std; int N; void f1() {     int x=5;     cout &lt;&lt; endl &lt;&lt; x;     P=2; //eroare } int P=9; int main () {     f1();     cout &lt;&lt; x; //eroare     P=7; //corect }</pre>                                                   | <pre># include &lt;iostream&gt; using namespace std; int N; void f1(int p) {     int x=p;     cout &lt;&lt; x; } int main () {     f1(77); }</pre>                                                                                                                             |
| <p><i>N este variabilă globală si poate fi accesată în cadrul oricărei funcții.</i></p> <p><i>x este variabilă locală, vizibilă doar în cadrul funcției f1()</i></p> <p><i>Se va afișa:</i></p> <p>4<br/>10<br/>5</p>              | <p><i>Compilerul generează eroare deoarece</i></p> <ul style="list-style-type: none"> <li><i>funcția main încearcă să acceseze variabila locală x care este vizibilă doar în funcția f1().</i></li> <li><i>variabila P este accesată în f1() înainte de a fi declarată.</i></li> </ul> | <p><i>Se afișează 77</i></p> <p><i>N este variabilă globală. Poate fi accesată în cadrul oricărei funcții.</i></p> <p><i>x este variabilă locală. Poate fi accesată doar în cadrul funcției f1()</i></p> <p><i>p este parametru formal. Poate fi accesat doar în f1().</i></p> |



## 5.4. Regula de omonimie

În cazul în care există o variabilă locală care are același nume cu o variabilă globală, aceste două variabile se numesc **variabile omonime**.

Variabilele locale sunt prioritare (ascund) variabilele globale omonime.

### Exemplul 5.4

```
# include <iostream>
using namespace std;
int N=10;
void f1()
{
    int N=2;
    cout << N<<" ";
}
int main ()
{
    f1();
    cout << N;
}
```

Variabila N este definită atât ca variabilă globală cât și ca variabilă locală în f1().

Se va afișa: 2 10

Funcția f1() acționează asupra variabilei locale N.

Funcția main() acționează asupra variabilei globale N.



**Întrebare.** Cum gestionează compilatorul cele două variabile omonime ?

**Răspuns:**

Variabilelor **globale** li se rezervă spațiu de memorie la începutul execuției programului, într-o zonă de memorie numită “zonă de date”. Acest spațiu va fi ocupat până la încheierea execuției programului.

Variabilelor **locale** li se rezervă spațiu într-o zonă specială de memorie numită “**stiva**”. La încheierea execuției subprogramului, conținutul stivei este eliberat. Din acest motiv, variabilele locale sunt vizibile doar în interiorul subprogramului în care au fost declarate.



## 6. Parametri formali și parametri actuali

**Parametri formali** apar în antetul subprogramului și sunt utilizați de subprogram pentru descrierea abstractă a unui proces de calcul .

**Parametri actuali** apar în instrucțiunea de apelare a unui subprogram și sunt folosiți la execuția unui proces de calcul pentru valori concrete.

Parametrii formali nu sunt variabile. O variabilă este caracterizată de nume, tip, și adresă. Legarea unui parametru formal la o **adresă** se realizează în timpul execuției instrucțiunii de apelare a subprogramului.