

## 4. Aplicații minimale C/C++ (Minimum C/C++ applications)

### 1. Obiective

- Înțelegerea structurii unui program C/C++
- Înțelegerea noțiunilor: comentarii, directive preprocesor, declarații globale, funcții, definiția și prototipul unei funcții, apelul unei funcții, parametri formali și parametri actuali
- Scrierea și testarea unor programe simple C/C++

### 1'. Objectives

- Understanding the structure of a C/C++ program
- Understanding the meaning of: comments, preprocessor directives, global declarations, functions, a function's definition and prototype, calling functions, formal and actual parameters
- Writing and testing some simple C/C++ programs

### 2. Breviar teoretic

Forma generală a unei aplicații C/C++ urmărește de obicei următoarele etape:

- Comentarii inițiale, ce prezintă scopul aplicației și realizatorul ei
- Directive preprocesor de tip *include*
- Directive preprocesor de tip *define*
- Declarații globale de variabile sau alte tipuri de date
- Prototipuri de funcții
- Funcția *main*
- Definirea celorlalte funcții din cadrul aplicației

Limbajul C/C++ este un limbaj procedural, la baza lui fiind procedura, numită *funcție*.

Tipurile de variabile în C sunt recunoscute prin cuvintele cheie:

<b><i>char</i></b> ,	pentru caractere;
<b><i>int</i></b> ,	pentru întregi cu semn;
<b><i>void</i></b> ,	neprecizat;
<b><i>float</i></b> ,	real simplă precizie;
<b><i>double</i></b> ,	real dublă precizie, iar modul de implementare poate fi modificat cu ajutorul declarațiilor suplimentare <i>signed, unsigned, long, short</i> .

În C++ avem în plus tipurile de date:

***bool*** – tipul Boolean, pentru stocarea valorilor booleene sau logice. O variabilă booleană poate avea valoarea *true* sau *false*  
***wchar\_t*** – tot un tip de date caracter, dar care este reprezentat pe 2 sau 4 octeți

Preprocesarea permite prelucrarea unui program sursă C sau C++ înainte de a fi supus compilării și asigură:

- includeri de fișiere cu text sursă;
- definiții și apeluri de macro-uri;
- compilare condiționată.

### 3. Exemple

Exemplele următoare, atât din acest capitol cât și din celelalte, au fost testate în mediul de programare MS Visual C++ 201y.

Specific acestui mediu de programare este utilizarea pentru intrări/ieșiri C++ a combinației:

```
#include <iostream>
using namespace std; // specifică utilizarea spațiului de nume standard
```

în loc de:

```
#include <iostream.h>
```

care se folosește în alte medii de programare C/C++.

**Exemplul 1:** program pentru citirea și afișarea unui întreg folosind funcții.

```
// directive preprocesor

// includerea unor fisiere antet ce contin prototipurile functiilor din
// biblioteca standard folosite in program
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>

// declaratii globale

// prototipuri functii: pentru fiecare functie precizeaza numele, lista cu
// tipurile parametrilor formali si tipul returnat
int cit_int(void);
void afis_int(int);

//functia main(): are aceeasi structura ca orice functie
// antet functie: nume, lista cu numele si tipurile parametrilor, tipul returnat

int main()
// corp functie
{
// declaratii locale
int n;                // variabila locala in care se va memora numarul citit

    // instructiuni
    n = cit_int();    // apel functie de citire si atribuire rezultat
    afis_int(n);
    _getch();        // apel functie de afisare
    return 0;
} //main

// definiții funcții

// functie pentru citirea unui intreg
// antet functie: nume, lista cu numele si tipul parametrilor, tipul returnat
int cit_int(void)
// corp functie
{
// declaratii locale
int nr;                // variabila intreaga in care se va memora valoarea citita

    // instructiuni
    // apel functie din biblioteca standard pentru afisarea unui mesaj
    printf("\nIntroduceti un numar intreg : ");
    // apel functie din biblioteca standard pentru citirea unui intreg
    scanf("%d", &nr);

    // returnare rezultat
    return nr;
} //cit_int

// functie pentru afisarea unui intreg
void afis_int(int nr)
```

```

{
    // apel functie din biblioteca standard pentru afisarea unui mesaj
    // si a numarului intreg primit ca paramteru
    printf("\nAti introdus numarul: %d\n", nr);
} // afis_int

```

**Exemplul 2:** program pentru calculul mediei aritmetice a două numere întregi.

```

// directive preprocesor
// includerea unor fisiere antet ce contin prototipurile functiilor din
// biblioteca standard folosite in program
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>

// declaratii globale

// prototipuri functii
int cit_int(void);
float medie_a(int, int);

int main()
{
    // declaratii locale
    float ma;           // variabila flotanta pentru media aritmetica
    int n1, n2;         // variabile intregi pentru numerele intregi ce vor fi citite

    // apel functie de citire a unui intreg
    n1 = cit_int();
    n2 = cit_int();

    // apel functie de calcul a mediei aritmetice
    ma = medie_a(n1, n2);

    // afisare rezultat
    printf("\n\t Media aritmetica : %f\n", ma);
    _getch();
    return 0;
}

// definitii functii

// functie de calcul a mediei aritmetice pentru doua numere intregi
float medie_a(int n1, int n2)
{
    // declaratii locale
    float medie;        // variabila flotanta pentru stocarea mediei aritmetice

    medie = (n1 + n2)/2.0;
    return medie;
}

// functie pentru citirea unui intreg
int cit_int(void)
{
    // declaratii locale

```

```
int nr;           // variabila întreaga în care se va memora valoarea citita

printf("\nIntroduceți un număr întreg : ");
scanf("%d", &nr);
return nr;
}
```

#### 4. Întrebări:

- Care este structura unui program C/C++ ?
- Ce sunt comentariile ? La ce se folosesc ?
- Care este deosebirea între prototipul și definiția unei funcții ?
- Cum se face apelul unei funcții ?
- Cum se face revenirea din funcții ?

#### 5. Teme:

1. Să se scrie un program pentru determinarea mediei aritmetice a trei numere neîntregi.
2. Să se scrie un program pentru determinarea mediei geometrice a două numere întregi.
3. Să se scrie un program C/C++ care definește o variabilă întreagă care va fi inițializată cu valori constante. Afișați rezultatul cu ajutorul supraîncărcării operatorului << și a lui *cout*.
4. Definiți un șir de caractere care va fi afișat cu *cout*. Definiți alte șiruri de caractere folosind secvențe escape. Verificați utilizarea spațiilor albe.
5. Să se scrie un program în care se dau 3 numere întregi și se cere să se calculeze suma lor ponderată, ponderile fiind numere subunitare a căror suma este 1.
6. Definiți într-un program constante simbolice de tipuri diferite (întregi, reale, șiruri de caractere). Afișați valorile acestor constante utilizând operatorul << și fluxul *cout*.
7. Definiți 3 numere reale *a*, *b*, și *c*. Afișați rezultatul operației  $1/a + 1/b + 1/c$ . Efectuați aceeași operație considerând ca și intrare numere întregi.

#### 4'. Questions

- What is the structure of a C/C++ program?
- What are comments? What are they used for?
- What is the difference between a function's prototype and definition?
- How can a function be called?
- How is a function's returning done?

#### 5. Homework

1. Write a program that determines the average value of 3 non-integer numbers.
2. Write a program that determines the geometric average of 2 integer numbers.
3. Write a C/C++ application that defines an integer variable, initialized with several constant values. Display its value by overloading the << operator and by using the *cout* object.
4. Define an array of characters that will be displayed using *cout*. Display other character arrays and use escape sequences. Verify the usage of the whitespaces.
5. Write a program that defines 3 integer values. Calculate and display their weighted sum, the weights being represented as positive values smaller than 1 that add up to 1.
6. Define several symbolic constants of different types (integer numbers, real numbers, arrays of characters). Display their values using *cout* and the << operator.
7. Define 3 real numbers named *a*, *b* and *c*. Display the value of  $1/a + 1/b + 1/c$ . Display the same result considering as input integer numbers.