

Αντικειμενοστραφείς Γλώσσες Προγραμματισμού C++ / ROOT

Ιωάννης Παπαδόπουλος

Τμήμα Φυσικής, Πανεπιστήμιο Ιωαννίνων

Σεπτέμβριος 2018

Περιεχόμενα

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- 1 Μεταγλωττισμένες γλώσσες προγραμματισμού / C++
 - Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++
 - Η διαδικασία μεταγλώττισης (compilation)
- 2 Γενικά για τη C++
- 3 Βασική σύνταξη της C++
 - hello.cpp
 - Στοιχεία σύνταξης
 - Χώροι ονομάτων
 - Εκφράσεις και εντολές
 - Τελεστές, τύποι, μεταβλητές
 - iostream: ρεύματα εισόδου/εξόδου

Περιεχόμενο του μαθήματος

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- 1° μέρος
 - Εισαγωγή στον προγραμματισμό σε C++
 - Βασικές έννοιες του αντικειμενοστραφούς προγραμματισμού
 - Εισαγωγή στη χρήση της Standard Template Library (STL)
- 2° μέρος
 - Εισαγωγή στη χρήση του λογισμικού ανάλυσης δεδομένων ROOT

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

Ένας **υπολογιστής**, αγνοώντας τα περιφερειακά του και εστιάζοντας στην *καρδιά* του, είναι απλά ένας **επεξεργαστής με κάποια μνήμη**, ικανός να **τρέχει στοιχειώδεις εντολές** όπως “πρόσθεσε τον αριθμό 3 στο περιεχόμενο της θέσης μνήμης 0x7fe2”.

Ένα απλό τετριμμένο πρόβλημα, π.χ. η λύση μίας πρωτοβάθμιας εξίσωσης, απαιτεί έναν μεγάλο αριθμό τέτοιων στοιχειωδών εντολών για τη λύση του.

Για το λόγο αυτό, προέκυψαν οι γλώσσες προγραμματισμού υψηλότερου επιπέδου, βάσει των οποίων ένα πρόγραμμα αναπτύσσεται ως ένα αρχείο κειμένου (πηγαίος κώδικας) που ακολουθεί τη σύνταξη της εκάστοτε γλώσσας προγραμματισμού.

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

Πλεονεκτήματα:

- **Συντομία:** οι γλώσσες προγραμματισμού μας επιτρέπουν να εκφράσουμε κοινές ακολουθίες εντολών πιο συνοπτικά. Η C++ παρέχει ιδιαίτερα ισχυρές συντομογραφίες.
- **Συντήρηση κώδικα:** η τροποποίηση ενός προγράμματος είναι εύκολη, απαιτώντας μερικές μόνο αλλαγές του κειμένου του πηγαίου κώδικα (οι οποίες αντιστοιχούν σε αλλαγή εκατοντάδων στοιχειωδών εντολών του επεξεργαστή). Η C++ όντας αντικειμενοστραφής, βοηθά περισσότερο στην ευκολία συντήρησης του κώδικα.

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- **Φορητότητα:** επεξεργαστές διαφορετική αρχιτεκτονικής, διαθέτουν διαφορετικά σύνολα στοιχειωδών εντολών. Τα προγράμματα κειμένου πηγαίου κώδικα, όπως της C++, μπορούν να μεταφραστούν σε σειρά στοιχειωδών εντολών για πολλούς διαφορετικούς επεξεργαστές. Η C++ είναι τόσο διαδεδομένη, ώστε μπορεί να χρησιμοποιηθεί για την συγγραφή προγραμμάτων για σχεδόν οποιοδήποτε επεξεργαστή.

Η C++ είναι μια γλώσσα υψηλού επιπέδου:

ένα πρόγραμμα γραμμένο σε C++ δεν ασχολείται με τις λεπτομέρειες των στοιχειωδών εντολών του επεξεργαστή, εξακολουθώντας όμως να παρέχει τη δυνατότητα πρόσβασης σε λειτουργίες χαμηλότερου επιπέδου (π.χ. διευθύνσεις μνήμης).

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

C++ / ROOT

Πρόγραμμα σε γλώσσα μηχανής για την πρόσθεση αριθμών που δίνονται στη γραμμή εντολής

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

```
1 ;--- This example was taken from http://asmtutor.com
2 ;--- It takes the command line arguments which should be integers,
3 ;--- adds them and displays their sum.
4 ;---
5 ;--- To compile and link:
6 ;--- nasm calc.asm -f elf ; ld -m elf_i386 calc.o -o calc.exe
7 ;--- To run:
8 ;--- ./calc.exe 10 20 1000 3000
9
10 ;-----
11 ; int slen(String message)
12 ; String length calculation function
13 slen:
14     push    ebx
15     mov     ebx, eax
16 nextchar:
17     cmp     byte [eax], 0
18     jz     finished
19     inc     eax
20     jmp     nextchar
21
22 finished:
23     sub     eax, ebx
24     pop     ebx
25     ret
```

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

Πρόγραμμα σε γλώσσα μηχανής για την πρόσθεση αριθμών που δίνονται στη γραμμή εντολής

```
26 ;-----  
27 ; void iprint(Integer number)  
28 ; Integer printing function (itoa)  
29 iprint:  
30  
31     push    eax           ; preserve eax on the stack to be restored after function runs  
32     push    ecx           ; preserve ecx on the stack to be restored after function runs  
33     push    edx           ; preserve edx on the stack to be restored after function runs  
34     push    esi           ; preserve esi on the stack to be restored after function runs  
35     mov     ecx, 0        ; counter of how many bytes we need to print in the end  
36  
37 divideLoop:  
38     inc     ecx           ; count each byte to print - number of characters  
39     mov     edx, 0        ; empty edx  
40     mov     esi, 10       ; mov 10 into esi  
41     idiv   esi           ; divide eax by esi  
42     add     edx, 48       ; convert edx to it's ascii representation  
43     push   edx           ; push edx (string representation of an integer) onto the stack  
44     cmp    eax, 0        ; can the integer be divided anymore?  
45     jnz    divideLoop    ; jump if not zero to the label divideLoop  
46  
47 printLoop:  
48     dec     ecx           ; count down each byte that we put on the stack  
49     mov     eax, esp      ; mov the stack pointer into eax for printing  
50     call   sprintf        ; call our string print function
```


Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

Πρόγραμμα σε γλώσσα μηχανής για την πρόσθεση αριθμών που δίνονται στη γραμμή εντολής

```
51     pop     eax           ; remove last character from the stack to move esp forward
52     cmp     ecx, 0       ; have we printed all bytes we pushed onto the stack?
53     jnz     printLoop   ; jump is not zero to the label printLoop

54     pop     esi           ; restore esi from the value we pushed onto the stack at the start
55     pop     edx           ; restore edx from the value we pushed onto the stack at the start
56     pop     ecx           ; restore ecx from the value we pushed onto the stack at the start
57     pop     eax           ; restore eax from the value we pushed onto the stack at the start
58     ret

59
60
61 ;-----
62 ; void sprint(String message)
63 ; String printing function
64 sprint:
65     push   edx
66     push   ecx
67     push   ebx
68     push   eax
69     call  slen
70
71     mov   edx, eax
72     pop   eax
73
74     mov   ecx, eax
75     mov   ebx, 1
```

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

C++ / ROOT

Πρόγραμμα σε γλώσσα μηχανής για την πρόσθεση αριθμών που δίνονται στη γραμμή εντολής

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```
    mov     eax, 4
    int     80h

    pop     ebx
    pop     ecx
    pop     edx
    ret

;-----
; void iprintLF(Integer number)
; Integer printing function with linefeed (itoa)
iprintLF:
    call    iprint           ; call our integer printing function

    push   eax               ; push eax onto the stack to preserve it
    mov    eax, 0Ah          ; move 0Ah into eax - 0Ah is the ascii character for a linefeed
    push   eax               ; push the linefeed onto the stack so we can get the address
    mov    eax, esp          ; move the address of the current stack pointer into eax for sprint
    call   sprint            ; call our sprint function
    pop    eax               ; remove our linefeed character from the stack
    pop    eax               ; restore the original value of eax before our function was called
    ret

;-----
; void exit()
```

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

C++ / ROOT

Πρόγραμμα σε γλώσσα μηχανής για την πρόσθεση αριθμών που δίνονται στη γραμμή εντολής

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προγραμματισμού

Γενικά για τη C++

```
101 ; Exit program and restore resources
102 quit:
103     mov     ebx, 0
104     mov     eax, 1
105     int     80h
106     ret
107
108 ;-----
109 ; int atoi(Integer number)
110 ; Ascii to integer function (atoi)
111 atoi:
112     push   ebx           ; preserve ebx on the stack to be restored after function runs
113     push   ecx           ; preserve ecx on the stack to be restored after function runs
114     push   edx           ; preserve edx on the stack to be restored after function runs
115     push   esi           ; preserve esi on the stack to be restored after function runs
116     mov     esi, eax      ; move pointer in eax into esi (our number to convert)
117     mov     eax, 0        ; initialise eax with decimal value 0
118     mov     ecx, 0        ; initialise ecx with decimal value 0
119
120 .multiplyLoop:
121     xor     ebx, ebx      ; resets both lower and upper bytes of ebx to be 0
122     mov     bl, [esi+ecx] ; move a single byte into ebx register's lower half
123     cmp     bl, 48        ; compare ebx register's lower half value against ascii 48 ('0')
124     jl     .finished     ; jump if less than to label finished
125     cmp     bl, 57        ; compare ebx register's lower half value against ascii 57 ('9')
```

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

C++ / ROOT

Πρόγραμμα σε γλώσσα μηχανής για την πρόσθεση αριθμών που δίνονται στη γραμμή εντολής

I. Παπαδόπουλος

```
126     jg     .finished      ; jump if greater than to label finished
127     cmp     bl, 10        ; compare ebx register's lower half value against ascii 10 (linefeed)
128     je     .finished      ; jump if equal to label finished
129     cmp     bl, 0         ; compare ebx register's lower half value against decimal 0 (end of string)
130     jz     .finished      ; jump if zero to label finished
131
132     sub     bl, 48         ; convert ebx register's lower half to decimal representation of ascii value
133     add     eax, ebx      ; add ebx to our interger value in eax
134     mov     ebx, 10       ; move decimal value 10 into ebx
135     mul     ebx           ; multiply eax by ebx to get place value
136     inc     ecx           ; increment ecx (our counter register)
137     jmp     .multiplyLoop ; continue multiply loop
138
139     .finished:
140     mov     ebx, 10       ; move decimal value 10 into ebx
141     div     ebx           ; divide eax by value in ebx (in this case 10)
142     pop     esi           ; restore esi from the value we pushed onto the stack at the start
143     pop     edx           ; restore edx from the value we pushed onto the stack at the start
144     pop     ecx           ; restore ecx from the value we pushed onto the stack at the start
145     pop     ebx           ; restore ebx from the value we pushed onto the stack at the start
146     ret
147
148     ; Calculator (ATOI)
149
150     SECTION .text
```

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα	151
Γλώσσες προγραμματισμού	152
Γενικά για τη C++	153
154	155
156	157
158	159
160	161
162	163
164	165
166	167
168	169
170	

Πρόγραμμα σε γλώσσα μηχανής για την πρόσθεση αριθμών που δίνονται στη γραμμή εντολής

```
global _start
_start:
    pop     ecx           ; first value on the stack is the number of arguments
    mov     edx, 0       ; initialise our data register to store additions

nextArg:
    cmp     ecx, 0h      ; check to see if we have any arguments left
    jz     noMoreArgs   ; if zero flag is set jump to noMoreArgs label (jumping over the loop end)
    pop     eax          ; pop the next argument off the stack
    call   atoi          ; convert our ascii string to decimal integer
    add     edx, eax     ; perform our addition logic
    dec     ecx          ; decrease ecx (number of arguments left) by 1
    jmp    nextArg      ; jump to nextArg label

noMoreArgs:
    mov     eax, edx     ; move our data result into eax for printing
    call   iprintLF     ; call our integer printing with linefeed function
    call   quit         ; call our quit function
```

Λόγος χρήσης γλωσσών προγραμματισμού όπως η C++

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

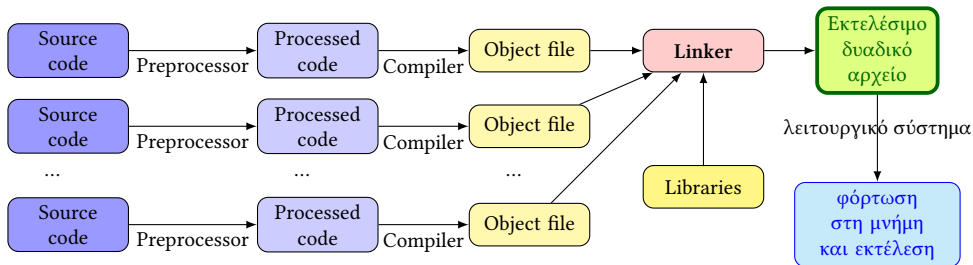
Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

Πρόγραμμα σε C++ για την πρόσθεση αριθμών που δίνονται στη γραμμή εντολής

```
1 //--- This C++ program takes the command line arguments which should be integers,  
2 //--- adds them and displays their sum.  
3 //--- To compile and link:  
4 //--- g++ -std=c++11 calc.cpp -o calc  
5 //--- To run:  
6 //--- ./calc 10 20 1000 3000  
7  
8 #include <iostream>           // for cout  
9 #include <string>            // for stod  
10  
11 using namespace std;  
12  
13 int main(int argc, char**argv) {  
14     double s=0;  
15     for (int i=1 ; i<argc ; i++) {  
16         s+=stod(argv[i]);  
17     }  
18     cout << "sum=" << s << endl;  
19     return 0;  
20 }
```

Η διαδικασία μεταγλώττισης (compilation)



- Object files: Πρόκειται για *ενδιάμεσα* δυαδικά αρχεία που περιέχουν στοιχώδεις εντολές που μπορεί να εκτελέσει ο επεξεργαστής (γλώσσα μηχανής), οι οποίες αντιστοιχούν στον πηγαίο κώδικα έπειτα από τη μεταγλώττιση (compilation).
- Τα επιμέρους object files συνδυάζονται από τον linker με τον απαραίτητο δυαδικό κώδικα που αναζητείται σε βιβλιοθήκες του συστήματος και παράγεται το τελικό εκτελέσιμο δυαδικό αρχείο.

Γενικά για τη C++

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- Η C++ είναι εξαιρετικά δημοφιλής, ιδιαίτερα για εφαρμογές που απαιτούν ταχύτητα ή / και πρόσβαση σε ορισμένες λειτουργίες χαμηλού επιπέδου.
- Δημιουργήθηκε το 1979 από τον Bjarne Stroustrup, αρχικά ως ένα σύνολο επεκτάσεων στη γλώσσα προγραμματισμού C.
- Η εκμάθησή της βασίζεται όχι τόσο στην ανάγνωση βιβλίων, αλλά στη συνεχή ενασχόληση με το σχεδιασμό (ανάλυση), τη συγγραφή και την αποσφαλμάτωση προγραμμάτων.
- Στο μάθημα αυτό, θα χρησιμοποιηθεί το πρότυπο **c++11** της γλώσσας, όπως τυποποιήθηκε το 2011.

Το πρώτο μου πρόγραμμα σε C++ ...

hello.cpp

```
1 // Το πρώτο μου πρόγραμμα σε C++...  
2 #include <iostream>  
3  
4 int main() {  
5     std::cout << "Hello World!!! Για σου Κόσμε..." << std::endl;  
6     return 0;  
7 }
```

Για τη δημιουργία εκτελέσιμου αρχείου:

```
g++ -std=c++11 hello.cpp -o hello.exe
```

Στοιχεία σύνταξης

είδος στοιχείου	περιγραφή	παραδείγματα
δεσμευμένες λέξεις	λέξεις με ειδικό για τον μεταγλωττιστή νόημα ονόματα για πράγματα που δεν είναι ενσωματωμένα στη C++	<code>int double for if auto const public private</code>
αναγνωριστικά	ενσωματωμένες στη C++ συγκεκριμένες τιμές που αναφέρονται άμεσα εντός του πηγαίου κώδικα για λογικές, μαθηματικές και άλλες πράξεις	<code>std cout x y il f_x</code>
σταθερές	σηματοδοτούν τη δομή του προγράμματος	<code>"Hello!" 24.3 1. -12 0x2f 'c'</code>
τελεστές	είδος κειμένου που ο μεταγλωττιστής θα αγνοήσει	<code>+ - && & % <<</code>
σημεία στίξης	κενά, στηλοθέτες, αλλαγή γραμμής, σχόλια	<code>{ } () , ;</code>
κενός χώρος		

Χρήση χώρων ονομάτων (namespaces)

C++ / ROOT

Ι. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

```
1 // Το πρώτο μου πρόγραμμα σε C++...  
2 #include <iostream>  
3  
4 int main() {  
5     std::cout << "Hello World!!! Γεια σου Κόσμε..." << std::endl;  
6     return 0;  
7 }
```

```
1 // Το πρώτο μου πρόγραμμα σε C++...  
2 #include <iostream>  
3 using namespace std;  
4 int main() {  
5     cout << "Hello World!!! Γεια σου Κόσμε..." << endl;  
6     return 0;  
7 }
```

εκφράσεις και εντολές

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- Οι **εκφράσεις (expressions)** αποτελούν παραστάσεις οι οποίες μπορούν να υπολογιστούν και να δώσουν ένα αποτέλεσμα. Μπορούν να αποτελούν τμήμα μίας εντολής.

```
(a==0) || (c!='A')  
y*z-3  
"Τμήμα φυσικής\n"
```

- Ένα πρόγραμμα αποτελείται από επιμέρους **εντολές (statements)**, δηλαδή μονάδες κώδικα που εκτελούν κάτι. Πρόκειται για τα βασικά δομικά στοιχεία του κώδικα. Τερματίζονται πάντοτε με τον χαρακτήρα ';'.

```
double x;  
std::cin >> x;  
cout << "Η τιμή του x είναι: " << y*z-3 << endl;  
if (x==0) cout << "δεν μπορώ να διαιρέσω με το x\n";
```

εκφράσεις και εντολές

- Μπορούν να οριστούν **μπλοκ εντολών** εντός αγκίστρων **{ }**

```
if (x==0) {  
    cout << "δεν μπορώ να διαιρέσω με το x\n";  
}  
else {  
    double y=(sqrt(x*x+1))/x;  
    cout << "Για x=" << x << " y=" << y << endl;  
}
```

```
for (int i=0; i<10 ; i++) {  
    int i2=i*i;  
    int i3=i*i*i;  
    cout << "i=" << i << " i2=" << i2 << " i3=" << i3 << endl;  
}
```

τελεστές

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- Για την εκτέλεση πράξεων, είναι διαθέσιμοι διάφοροι τελεστές.
 - αριθμητικοί: + - * / % ++ -- += -= *= /= %=
 - λογικοί: ! && ||
 - δυαδικοί (bitwise): << >> ~ & | ^
 - επιλογής μέλους: . ->

βασικοί τύποι δεδομένων

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- Κάθε έκφραση έχει έναν **τύπο**, ο οποίος περιγράφει το είδος των δεδομένων στο οποίο αναφέρεται η τιμή της. Π.χ.
 - 0 100 -3 → ακέραιοι
 - 1.5 2. 21.123 -3.2 → δεκαδικοί (κινητής υποδιαστολής)
 - 'q' 'θ' '>' → χαρακτήρες
 - "Καλημέρα... \n" → αλφαριθμητική σειρά

βασικοί τύποι δεδομένων

τύπος	μέγεθος ¹ (bytes)	περιγραφή	πεδίο τιμών
bool	1	ακέραιος	0 (ή false), 1 (ή true)
char	1	ακέραιος	-128 – 127 ή 0 – 255
unsigned char	1	ακέραιος	0 – 255
signed char	1	ακέραιος	-128 – 127
int	4	ακέραιος	-2147483648 – 2147483647
unsigned int	4	ακέραιος	0 – 4294967295
signed int	4	ακέραιος	-2147483648 – 2147483647
float	4	δεκαδικός με 7 σημαντικά ψηφία	$\pm 3.4 \cdot 10^{\pm 38}$, 0, nan, inf, -inf
double	8	δεκαδικός με 15 σημαντικά ψηφία	$\pm 1.7 \cdot 10^{\pm 308}$, 0, nan, inf, -inf

¹Το μέγεθος και κατά συνέπεια το πεδίο τιμών δεν είναι απολύτως καθιερωμένα. Εξαρτώνται από την έκδοση και την υπολογιστική πλατφόρμα.

πράξεις και τύποι δεδομένων

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- Πράξεις μεταξύ τιμών του ίδιου τύπου δίνουν συνήθως αποτέλεσμα του ίδιου τύπου, π.χ.:

`int / int → int (3/5→0)`

`float - float → float`

`double * double → double`

- Πράξεις μεταξύ τιμών διαφορετικού τύπου δίνουν αποτέλεσμα του “ισχυρότερου” τύπου, π.χ.:

`int / float → float (3/5.→0.6)`

`int + double → double`

`float * double → double`

μεταβλητές

- Οι **μεταβλητές (variables)** χρησιμοποιούνται για να δώσουμε όνομα σε κάποια τιμή που αποθηκεύεται στη μνήμη, ώστε να μπορέσουμε να αναφερθούμε αργότερα σε αυτή.
- Τα ονόματά τους δεν μπορούν να ξεκινούν από ψηφία, και αποτελούνται από τους λατινικούς χαρακτήρες (a-z, A-Z), τα ψηφία (0-9) και τον χαρακτήρα ‘_’ (underscore), π.χ.:
f, x, y1, y2, f_x, Mass, time1, w_1, _My_Var
- Δηλώνονται με συγκεκριμένο τύπο, και λαμβάνουν τιμή εν γένει μέσω ανάθεσης (assignment) με τον τελεστή ‘=’, π.χ.:

```
int i=100; // δήλωση και αρχικοποίηση σε τιμή 100 με χρήση =
int j(3); // δήλωση και αρχικοποίηση σε τιμή 3 με χρήση ()
int k{234}; // δήλωση και αρχικοποίηση σε τιμή 234 με χρήση {}
i=2*i; // ανάθεση της τιμής που προκύπτει από τον υπολογισμό
double x; // δήλωση
x=5+i/30.; // ανάθεση της τιμής που προκύπτει από τον υπολογισμό
cout << "i=" << i << " x=" << x << endl;
```

μεταβλητές

- Εμβέλεια (scope): Οι μεταβλητές μπορούν να δηλωθούν οπουδήποτε μέσα στον κώδικα, και έχουν τοπική εμβέλεια (από το σημείο δήλωσης και κάτω, και πάντα εντός του μπλοκ όπου δηλώνονται).

```
for (int i=0 ; i<10 ; i++) {  
    cout << "i2=" << i*i << endl;  
    cout << "i3=" << i*i*i << endl;  
}  
cout << "i=" << i << endl; // ΛΑΘΟΣ: το i δεν υφίσταται εδώ...
```

```
double f1(double x) { // Τα x και y έχουν ισχύ μόνον εντός της συνάρτησης f1  
    double y=x*x-3; // και δεν έχουν καμία σχέση με αυτά της f2  
    return y; // Πρόκειται για συνωμυμία...  
}
```

```
double f2(double x) { // Τα x και y έχουν ισχύ μόνον εντός της συνάρτησης f2  
    double y=x+1; // και δεν έχουν καμία σχέση με αυτά της f1  
    return y; // Πρόκειται για συνωμυμία...  
}
```

υπονοούμενοι τύποι μεταβλητών (auto και decltype)

- Κατά την αρχικοποίηση μίας μεταβλητής, ο μεταγλωττιστής μπορεί να καταλάβει και να θέσει τον τύπο της. Αυτό μπορεί να γίνει χρησιμοποιώντας τη λέξη **auto** αντί συγκεκριμένου τύπου, π.χ.

```
int foo = 0;  
auto bar = foo; // ισοδύναμο με: int bar = foo;
```

όπου το `bar` λαμβάνει τον τύπο του `foo` με το οποίο αρχικοποιείται.

- Είναι δυνατόν να υπονοηθεί ο τύπος μίας μεταβλητής η οποία δεν αρχικοποιείται, χρησιμοποιώντας τη λέξη **decltype** αντί συγκεκριμένου τύπου, όπως στο ακόλουθο παράδειγμα όπου το `bar` λαμβάνει τον τύπο του `foo`.

```
int foo = 0;  
decltype(foo) bar; // ισοδύναμο με: int bar;
```

- Χρήση: όταν ο τύπος δεν γίνεται να προσδιοριστεί διαφορετικά ή όταν αυξάνουν την ευκολία ανάγνωσης του πηγαίου κώδικα ².

²Τα παραδείγματα είναι μάλλον ατυχή για την επίδειξη της χρησιμότητας των `auto` και `decltype`... 22/27

Ο τελεστής εξόδου: <<

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- Έννοια του ρεύματος χαρακτήρων: Τα δεδομένα οποιουδήποτε τύπου μετατρέπονται σε χαρακτήρες και μεταδίδονται σε ένα ρεύμα (stream) χαρακτήρων.
- Ο τελεστής εξόδου << χρησιμοποιείται για την αποστολή τέτοιων δεδομένων σε ένα ρεύμα. Για παράδειγμα, η εντολή:
$$a \ll b;$$
σημαίνει “Στείλε στο ρεύμα **a** την τιμή του **b** ως ρεύμα χαρακτήρων”.
- Βασικά ρεύματα εξόδου (διαθέσιμα μέσω της επικεφαλίδας <iostream>):
 - cout** καθιερωμένο ρεύμα εξόδου
 - clog** καθιερωμένο ρεύμα εξόδου για μηνύματα σφαλμάτων
 - cerr** καθιερωμένο ρεύμα εξόδου για μηνύματα σφαλμάτων χωρίς προσωρινή αποθήκευση (buffer)

Ο τελεστής εξόδου: <<

- Παράδειγμα:

```
int i=3;
cout << "i=";
cout << i;
cout << " i2=";
cout << i*i;
cout << " i3=";
cout << i*i*i;
cout << endl;
```

i=3 i²=9 i³=27

- Ο τελεστής << μπορεί να χρησιμοποιηθεί πολλές φορές εντός μίας εντολής. Το παραπάνω παράδειγμα μπορεί να γραφεί πιο σύντομα ως:

```
int i=3;
cout << "i=" << i << " i2=" << i*i << " i3=" << i*i*i << endl;
```

i=3 i²=9 i³=27

endl αποστολή τέλους γραμμής ('\n') και άδειασμα του buffer (flush)

Ο τελεστής εισόδου: >>

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- Ο τελεστής εισόδου >> χρησιμοποιείται για την ανάγνωση δεδομένων από ένα ρεύμα. Για παράδειγμα, η εντολή:

```
a >> b;
```

σημαίνει “Διάβασε χαρακτήρες από το ρεύμα **a** και θέσε τους ως τιμή του **b**”.

- Βασικό ρεύμα εισόδου (διαθέσιμο μέσω της επικεφαλίδας <iostream>):
`cin` καθιερωμένο ρεύμα εισόδου

Ο τελεστής εισόδου: >>

C++ / ROOT

I. Παπαδόπουλος

Περιεχόμενα

Γλώσσες προ-
γραμματισμού

Γενικά για τη
C++

- Παράδειγμα³:

```
#include <iostream>
using namespace std;
int main () {
    int i;
    cout << "Δώσε έναν ακέραιο: ";
    cin >> i;
    cout << "Η τιμή που έδωσες είναι: " << i << endl;
    return 0;
}
```

Δώσε έναν ακέραιο: 123

Η τιμή που έδωσες είναι: 123

³Εκτέλεση στο https://www.onlinegdb.com/online_c++_compiler

Ο τελεστής εισόδου: >>

- Παράδειγμα⁴:

```
#include <iostream>
#include <string>
using namespace std;
int main () {
    string ono, epw;
    int am;
    cout << "Δώσε όνομα επώνυμο και AM: ";
    cin >> ono >> epw >> am ;
    cout << "Ονομάζεσαι " << ono << " " << epw
        << " και έχεις AM=" << am << endl;
    return 0;
}
```

Δώσε όνομα επώνυμο και AM: Ιωάννης Παπαδόπουλος 12345
Ονομάζεσαι Ιωάννης Παπαδόπουλος και έχεις AM=12345

⁴Εκτέλεση στο https://www.onlinegdb.com/online_c++_compiler