Εισαγωγή στον Προγραμματισμό

Ενότητα C++ (75%) :

4η εβδομάδα

Παναγιώτης Τζουνάκης

(συμπεριλαμβάνει υλικό που ευγενικά προσέφερε ο καθηγητής κ. Γεώργιος Ραχώνης)

Χειμώνας 2026



Τμήμα Μαθηματικών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



Βιβλιογραφία για C++

Βιβλιογραφία μαθήματος (Εύδοξος)

1. Βιβλίο [122077282]: C++: Από τη Θεωρία στην Εφαρμογή, Γ. Τσελίκης
2. Βιβλίο [13598]: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ ΤΗ C++, ROBERT LAFORE
3. Βιβλίο [50655981]: C++, 9η Έκδοση, Savitch Walter

Επιπρόσθετη βιβλιογραφία για μελέτη

- Walter Savitch, Problem Solving with C++, Addison-Wesley (10th edition) 2018.
- David Harel, Yishai Feldman, Algorithmics, The Spirit of Computing, Addison-Wesley (3rd edition) 2004.
- T. H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithm", MIT Press (3rd edition) 2009.
- Bjarne Stroustrup, Η Γλώσσα Προγραμματισμού C++, 4η εκδ., Εκδόσεις Κλειδάριθμος 2014.
- Αναλυτικές σημειώσεις του κ. Γ. Τσελίκη, στην ιστοσελίδα του μαθήματος https://clab.math.auth.gr/sites/default/files/2025-03/CplusplusTselikis_Chapter_01_Chapter_26.pdf , κατόπιν αδείας του συγγραφέα
- Δημοσιεύσεις από τον ίδιο το δημιουργό: https://www.stroustrup.com/papers.html

Last resort: (e.g. google) search for "C++ tutorial" or "C++ tutorial greek" or "C++ εγχειρίδιο"

Ενδεικτικές διαδικτυακές πηγές για προγραμματισμό με C, C++

- https://ocw.uc3m.es/course/view.php?id=213
- http://greenteapress.com/thinkcpp/thinkCScpp.pdf
- https://rooksguide.files.wordpress.com/2013/12/rooks-guide-isbn-version.pdf
- https://www.linuxlinks.com/excellent-free-books-learn-c-plus-plus/
- https://www.ece.uvic.ca/~frodo/cppbook/ (textbook-exercises, slides, SDE, videos, etc.)
- https://www.e-booksdirectory.com/details.php?ebook=2518
- https://www.materials.uoc.gr/el/undergrad/courses/ETY215/notes.pdf (στα **Ελληνικά**!)

C++ (αλλιώς CPP ή CXX)



- 1979 Bjarne Stroustrup
  "C with Classes"
- 1983 1η διανομή εντός AT&T
- 1985 πρώτη εμπορική έκδοση

https://www.stroustrup.com/bs_faq.html

περισσότερα: π.χ.
https://en.wikipedia.org/wiki/C++

C++

Logo endorsed by the C++ standards committee

Paradigms	Multi-paradigm: procedural, imperative, functional, object-oriented, generic, modular, reflective[1], Design by contract
Family	C
Designed by	Bjarne Stroustrup
Developer	ISO/IEC JTC 1 (Joint Technical Committee 1) / SC 22 (Subcommittee 22) / WG 21 (Working Group 21)
First appeared	1985; 41 years ago
Stable release	C++23 (ISO/IEC 14882:2024) / 19 October 2024; 2 years ago
Preview release	C++26 (N5032)[2] / 15 December 2025; 10 months ago
Typing discipline	Static, strong, nominative, partially inferred
OS	Cross-platform
Filename extensions	.C, .cc, .cpp, .cxx, .c++, .h, .H, .hh, .hpp, .hxx, .h++ .cppm, .ixx[3]
Website	isocpp.org

Major implementations

GCC, LLVM Clang, Microsoft Visual C++, Embarcadero C++Builder, Intel C++ Compiler, IBM XL C++, EDG

Influenced by

Ada, ALGOL 68,[4] BCPL,[5] C[a], CLU,[4] F#,[6] [b] ML, Mesa,[4] Modula-2,[4] Simula, Smalltalk[4]

Influenced

Ada 95, C#,[7] C99, Carbon, Chapel,[8] Clojure, [9] D, Java,[10] JS++,[11] Lua,[12] Nim,[13] Objective-C++, Perl, PHP, Python,[14] Rust[15]

C++ Programming at Wikibooks

Το αλφάβητο της C++

- Γράμματα: Κεφαλαία και πεζά Αγγλικά γράμματα

  A,B,…,Z,a,b,…,z

- Ψηφία 0,1,2,3,4,5,6,7,8,9

- Ειδικοί χαρακτήρες

  ! # $ % & ‘ ( ) * + - / < > : ;

  Κενό . = \ [ ] ^ { } ? “ |

Το λεξιλόγιο της C++

Το λεξιλόγιο της C++ περιλαμβάνει 8 κατηγορίες:

1. ***ονόματα (identifiers),***
2. ***αριθμούς (numbers),***
3. ***χαρακτήρες (characters),***
4. ***αλυσίδες χαρακτήρων (strings),***
5. ***τελεστές (operators),***
6. ***διαχωριστές (separators ή punctuators),***
7. ***λέξεις κλειδιά (keywords) και***
8. ***σχόλια (comments).***

Βλ. επίσης https://www.stroustrup.com/glossary.html





Ονόματα

- Περιέχουν 1-(32) αλφαριθμητικούς χαρακτήρες (τα 26 γράμματα της αγγλικής αλφαβήτου, τα 10 ψηφία και το σύμβολο _).
- Ξεκινούν πάντα με αγγλικό γράμμα ή το χαρακτήρα υπογράμμισης.
- Κεφαλαία και πεζά γράμματα θεωρούνται διαφορετικά.
- Δεν επιτρέπεται η χρήση ονομάτων που έχουν ειδική σημασία για τη C++ (λέξεις «*κλειδιά*»).

Τι δεν επιτρέπεται

- Δεν επιτρέπονται σύμβολα εκτός από το «_».
- Δεν επιτρέπονται τα κενά.
- Δεν επιτρέπονται οι Ελληνικοί χαρακτήρες.

Παραδείγματα

Επιτρεπτά ονόματα

AF1, DAY, Day, FATHER_NAME

Μη επιτρεπτά ονόματα

1DC, ΩΡΑ, FATHER-NAME





Αριθμοί

- ***Ακέραιοι αριθμοί.***

  [-32768, 32767] (απαιτούνται 16 bits)

  [-2147483648, 2147483647] (32 bits)

  [0, 65535] (ακέραιοι χωρίς πρόσημο)

  [0, 2^{32}-1]

- ***Πραγματικοί αριθμοί.***

Απλής ακριβείας

$[-3.4\text{x}10^{38}, \ -3.4\text{x}10^{-38}] \ [3.4\text{x}10^{-38}, 3.4\text{x}10^{38}]$

πλήθος *σημαντικών* ψηφίων: 7

Διπλής ακριβείας

$[-1.7\text{x}10^{308}, -1.7\text{x}10^{-308}] \ [1.7\text{x}10^{-308}, 1.7\text{x}10^{308}]$

πλήθος *σημαντικών ψηφίων*: 15

Διπλής ακριβείας

$[-1.7\text{x}10^{4932}, -1.7\text{x}10^{-4932}] \ [1.7\text{x}10^{-4932}, 1.7\text{x}10^{4932}]$

πλήθος *σημαντικών ψηφίων*: 19





Οι πραγματικοί αριθμοί χωρίζονται σε δύο κατηγορίες :

- ***Πραγματικοί αριθμοί σταθερής υποδιαστολής*** (θετικοί ή αρνητικοί αριθμοί που περιέχουν υποδιαστολή) π.χ. 34.5

(https://en.wikipedia.org/wiki/Fixed-point_arithmetic)

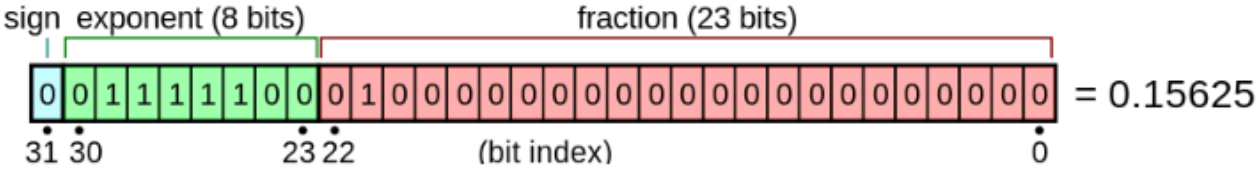
- ***Πραγματικοί αριθμοί κινητής υποδιαστολής*** (θετικοί ή αρνητικοί αριθμοί που είναι εκφρασμένοι σε εκθετική μορφή *xxxxE±xxxx* όπου η βάση είναι θετικός ή αρνητικός ακέραιος ή δεκαδικός αριθμός, ενώ ο εκθέτης είναι θετικός ή αρνητικός ακέραιος αριθμός) π.χ.
- Ο αριθμός 0.345E+2 συμβολίζει τον 34.5
- Ο αριθμός 0.345E-2 συμβολίζει τον 0.00345

(https://en.wikipedia.org/wiki/Floating-point_arithmetic)





IEEE 754 single-precision binary floating-point format: binary32



The real value assumed by a given 32-bit *binary32* data with a given *sign*, biased exponent *e* (the 8-bit unsigned integer), and a *23-bit fraction* is

$$(-1)^{b_{31}} \times 2^{(b_{30}b_{29}\ldots b_{23})_2 - 127} \times (1.b_{22}b_{21}\ldots b_0)_2 ,$$

which yields

bias explicit bit

$$\text{value} = (-1)^{\text{sign}} \times 2^{(E-127)} \times \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i}\right) .$$

In this example:

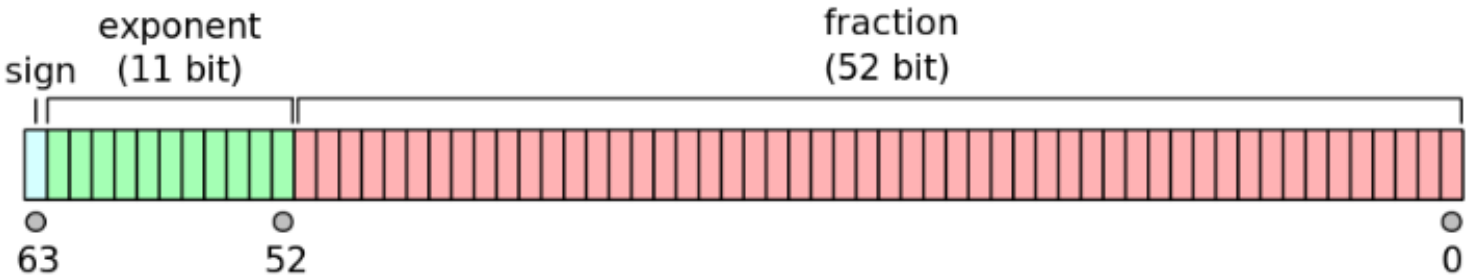
- $\text{sign} = b_{31} = 0$,
- $(-1)^{\text{sign}} = (-1)^0 = +1 \in \{-1, +1\}$,
- $E = b_{30}b_{29}\ldots b_{23} = \sum_{i=0}^{7} b_{23+i} 2^{+i} = 124 \in \{1, \ldots, (2^8 - 1) - 1\} = \{1, \ldots, 254\}$,
- $2^{(E-127)} = 2^{124-127} = 2^{-3} \in \{2^{-126}, \ldots, 2^{127}\}$,
- $1.b_{22}b_{21}\ldots b_0 = 1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} = 1 + 1 \cdot 2^{-2} = 1.25 \in \{1, 1 + 2^{-23}, \ldots, 2 - 2^{-23}\} \subset [1; 2 - 2^{-23}] \subset [1; 2)$.

thus:

- $\text{value} = (+1) \times 2^{-3} \times 1.25 = +0.15625$.

https://en.wikipedia.org/wiki/Single-precision_floating-point_format

IEEE 754 double-precision binary floating-point format: binary64



The real value assumed by a given 64-bit double-precision datum with a given biased exponent e and a 52-bit fraction is

$$(-1)^{\text{sign}}(1.b_{51}b_{50}\ldots b_0)_2 \times 2^{e-1023}$$

or

$$(-1)^{\text{sign}}\left(1+\sum_{i=1}^{52} b_{52-i}2^{-i}\right) \times 2^{e-1023}$$

https://en.wikipedia.org/wiki/Double-precision_floating-point_format

Χαρακτήρες

- Όλοι οι χαρακτήρες που υποστηρίζονται από τον Η/Υ μέσα σε αποστρόφους π.χ. ‘a’, ‘γ’. Ισχύουν τα εξής :
  - Για να δηλώσουμε τον χαρακτήρα απόστροφο πρέπει να τον γράψουμε μετά από το \ δηλαδή \’.
  - Ο χαρακτήρας ‘Α’ είναι διαφορετικός από το χαρακτήρα ‘a’.





- Ο χαρακτήρας ‘7’ διαφέρει από τον αριθμό 7. Ο πρώτος δεν μπορεί να πάρει μέρος σε αριθμητικές πράξεις ενώ ο δεύτερος μπορεί.
- Κάποιοι χαρακτήρες διακρίνονται πιο δύσκολα… Πρέπει να δοθεί προσοχή στο διαχωρισμό του μηδέν 0 και του κεφαλαίου όμικρον Ο, όπως και του 1 (ένα) από το l (ℓ δηλ. το λατινικό γράμμα el).

Αλυσίδες χαρακτήρων

Πρόκειται για ακολουθία χαρακτήρων μεταξύ διπλών αποστρόφων π.χ. “Μαθηματικά”, “123”.

Προσέξτε ότι το ‘α’ είναι διαφορετικό από το “α”.





Τελεστές

Ο τελεστής είναι σύμβολο που δηλώνει μια πράξη μεταξύ τελεστέων (όρων της πράξης), π.χ. αριθμών, αλυσίδων χαρακτήρων κ.λπ.

Υπάρχουν 3 κατηγορίες τελεστών:

- αριθμητικοί,
- σύγκρισης και
- λογικοί :





- ***Τελεστές αριθμητικοί***

  Χρησιμοποιούνται για πράξεις μεταξύ αριθμών.

Τελεστής	**Λειτουργία**	**Σύνταξη**
+	Πρόσθεση	X+Y
-	Αφαίρεση	X-Y
*	Πολλαπλασιασμός	X*Y
/	Διαίρεση	X/Y
%	Υπόλοιπο ακέραιας διαίρεσης	X%Y

- ***Τελεστές Σύγκρισης***

Χρησιμοποιούνται για σύγκριση μεταξύ αριθμών (με τη συνήθη μαθηματική διάταξη), αλυσίδων χαρακτήρων (λεξικογραφική διάταξη)

Τελεστής	**Λειτουργία**	**Σύνταξη**
==	Ισότητα	X==Y
!=	Ανισότητα	X!=Y
>	Μεγαλύτερο	X>Y
<	Μικρότερο	X<Y
>=	Μεγαλύτερο ή ίσο	X>=Y
<=	Μικρότερο ή ίσο	X<=Y





- ***Τελεστές Λογικοί***

  Χρησιμοποιούνται για την εκτέλεση λογικών πράξεων.

Τελεστής	**Λειτουργία**
!	Λογική άρνηση
&&	Σύζευξη
||	Διάζευξη

Οι πίνακες αληθείας των λογικών τελεστών είναι :

X	Y	X&&Y	X||Y	!X
1	1	1	1	0
1	0	0	1	0
0	1	0	1	1
0	0	0	0	1

Η λογική τιμή (σταθερά) “false” αντιστοιχεί στην αριθμητική τιμή (σταθερά) 0.
Η λογική τιμή (σταθερά) “true” αντιστοιχεί στην αριθμητική τιμή (σταθερά) 1.
Σημείωση: προκειμένου για μετατροπή από αριθμητικό σε λογικό τύπο δεδομένων,

1. η τιμή 0 μετατρέπεται σε false,
2. κάθε τιμή διαφορετική από 0, μετατρέπεται σε true.

βλ.
https://en.cppreference.com/cpp/language/implicit_conversion#Boolean_conversions





Διαχωριστές (separators ή punctuators)

Πρόκειται για χαρακτήρες με συγκεκριμένη λειτουργία στη C++. Λειτουργούν ως σημεία στίξης. Οι κυριότεροι είναι :

= Καταχώριση

( ) Παρενθέσεις

; Δήλωση τέλους εντολής





Λέξεις - Κλειδιά

Πρόκειται για λέξεις με ειδική σημασία στη C++:

asm	**auto**	**break**	**case**
catch	**char**	**class**	**const**
continue	**default**	**delete**	**do**
double	**else**	**enum**	**extern**
float	**for**	**friend**	**goto**
if	**inline**	**int**	**long**
new	**operator**	**private**	**protected**
public	**register**	**return**	**short**
signed	**sizeof**	**static**	**struct**
switch	**template**	**this**	**throw**
try	**typedef**	**union**	**unsigned**
virtual	**void**	**violate**	**while**





Σχόλια

Οποιοιδήποτε χαρακτήρες ακολουθούν το σύμβολο // (μέχρι το τέλος της γραμμής) θεωρούνται σχόλια και δεν μεταφράζονται από τον compiler της C++.

***Παράδειγμα*:**

```
// Αυτό είναι ένα (σύντομο) σχόλιο
```

***Άλλο Παράδειγμα*:**

```
/* Αυτό είναι ένα σχόλιο
   που πιάνει πολλές
   γραμμές! */
```





Κατηγορίες δεδομένων

Μεταβλητές. Δεδομένα που η τιμή τους μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος.

Πιο συγκεκριμένα, με τον όρο ***μεταβλητή*** εννοούμε μια (ή παραπάνω) θέση η οποία δημιουργείται στη μνήμη του Η/Υ, για να δεχτεί ένα συγκεκριμένο τύπο δεδομένων, και η οποία έχει ένα χαρακτηριστικό όνομα που πληροί τους κανόνες που θέσαμε προηγουμένως.





Τύποι δεδομένων

Όνομα	Σύνολο τιμών
int	ακέραιοι στο διάστημα [-32768, 32767] (μήκος 16bits, σε 32-bit αρχιτεκτονικές)
long int	ακέραιοι στο διάστημα [-2147483648, 2147483647] μήκος 32bits
unsigned int	ακέραιοι στο διάστημα [0, 65535]
unsigned long	ακέραιοι στο διάστημα $[0, 2^{32}-1]$





float $[-3.4 \times 10^{38}, -3.4 \times 10^{-38}]$ $[3.4 \times 10^{-38}, 3.4 \times 10^{38}]$

double $[-1.7 \times 10^{308}, -1.7 \times 10^{-308}]$ $[1.7 \times 10^{-308}, 1.7 \times 10^{308}]$

long double

$[-1.7 \times 10^{4932}, -1.7 \times 10^{-4932}]$ $[1.7 \times 10^{-4932}, 1.7 \times 10^{4932}]$

char χαρακτήρες ή ακέραιοι στο διάστημα -128 έως 127

αναλυτικότερα: π.χ.

https://en.cppreference.com/w/cpp/language/types

Εκφράσεις στη C++

Τύπος	Τελεστής	Σειρά προτεραιότητας για τελεστές με την ίδια προτεραιότητα
Αριθμητικός	pow(x,y)	
	* / %	Αριστερά προς Δεξιά
	+ -	Α προς Δ
Σύγκρισης	> >=	Α προς Δ
	< <=	Α προς Δ
	== !=	Α προς Δ
Λογικοί	!	**Δ προς Α**
	&& ||	Α προς Δ

Μερικές παρατηρήσεις για για τις παραστάσεις στην C++

- Προσοχή στις πράξεις που έχουν ίδια προτεραιότητα, π.χ. *η παράσταση a/b/c/d είναι ισοδύναμη με την ((a/b)/c)/d.*
- Infix notation για δυαδικούς τελεστές, δηλ. ο τελεστής ανάμεσα στους τελεστέους, π.χ. 3 * 4
- Δεν μπορούμε να βάλουμε δύο αριθμητικούς τελεστές τον ένα δίπλα στον άλλο, π.χ. 3*-4, πρέπει να χρησιμοποιούμε παρενθέσεις 3*(-4).
- Οπουδήποτε έχετε αμφιβολία για την σειρά εκτέλεσης των πράξεων **χρησιμοποιείστε παρενθέσεις**





Μαθηματικές συναρτήσεις στη C++

Όνομα	Περιγραφή
fabs(x)	Απόλυτη τιμή του x
sqrt(x)	Τετραγωνική ρίζα του x
sin(x)	Ημίτονο του x σε ακτίνια
asin(x)	Τόξο ημίτονου του x
cos(x)	Συνημίτονο του x σε ακτίνια
acos(x)	Τόξο συνημίτονου του x
tan(x)	Εφαπτομένη του x σε ακτίνια
atan(x)	Τόξο εφαπτομένης του x
exp(x)	e^x





log(x)	**Φυσικός λογάριθμος του x**
log10(x)	**Δεκαδικός λογάριθμος του x**
sinh(x)	**Υπερβολικό ημίτονο του x**
cosh(x)	**Υπερβολικό συνημίτονο του x**
tanh(x)	**Υπερβολική εφαπτομένη του x**
pow(x,y)	x^y





“Hello World (again!)”

```
// my first C++ program
#include <iostream>


int main() {
        std::cout << "Hello World!";
        std::cout << "\nJust printed a(n invisible)
newline character!";
        return 0;
}
```





Ασκήσεις

Να μετατρέψετε τις παρακάτω αριθμητικές εκφράσεις, σε μορφή αποδεκτή από τη γλώσσα C++.

Ελέγξτε την ορθότητα της έκφρασης ενθέτοντας την μέσα στο πρόγραμμα “Hello World” (με ένα ‘;’ στο τέλος της γραμμής) πριν δείτε τη λύση…





$$\frac{x-\eta\mu^2(x)}{\varepsilon\phi^2(x-1)+1}$$

$$\frac{x-\eta\mu^2(x)}{\varepsilon\phi^2(x-1)+1}$$

(x-pow(sin(x),2))/(pow(tan(x-1),2)+1)





“Hello World (again!)” E1

```
// my first C++ program with an expression
#include <iostream>
#include <math.h>
#define x 0.5
int main() {
        (x-pow(sin(x),2))/(pow(tan(x-1),2)+1);
        std::cout << "Hello World!\n";
        return 0;
}
```





$$\left| \frac{1+\sqrt{x-2}}{x^2-1} \right|$$

$$\left| \frac{1+\sqrt{x-2}}{x^2-1} \right|$$

```
fabs((1+sqrt(x-2))/(pow(x,2)-1))
```

“Hello World (again!)” E2

```
// my first C++ program with an expression
#include <iostream>
#include <math.h>
#define x 0.5
int main() {
        fabs((1+sqrt(x-2))/(pow(x,2)-1));
        std::cout << "Hello World!\n";
        return 0;
}
```





$$e^{\log(x)} - \frac{y}{x^3}$$

$$e^{\log(x)} - \frac{y}{x^3}$$

exp(log10(x))-y/pow(x,3)

“Hello World (again!)” E3

```
// my first C++ program with an expression
#include <iostream>
#include <math.h>
#define x 0.5
#define y 0.6
int main() {
        exp(log10(x))-y/pow(x,3); //useless?
        std::cout << "Hello World!\n";
        return 0;
}
```





$$|x - y| \leq 1$$

$$|x - y| \leq 1$$

fabs(x-y)<=1

$$-1 \leq \varepsilon\phi(x) \leq 3$$

$$-1 \leq \varepsilon\phi(x) \leq 3$$

(tan(x)>=-1)&&(tan(x)<=3)

Δομή προγράμματος στη C++

```
#include <header file>
using namespace std;
Πρωτότυπο συνάρτησης (function definition)
int main( )
{
    Δηλώσεις μεταβλητών
    Εκτελέσιμες εντολές
    return 0;
 }
    Δήλωση συνάρτησης (function declaration)
{
    Δηλώσεις μεταβλητών
     Εκτελέσιμες εντολές
 }
```





Η εντολή #include ζητάει από τον μεταγλωττιστή να ενσωματώσει ένα αρχείο (το header file) από την βιβλιοθήκη της C++ στο πρόγραμμά μας, π.χ.

```
#include <iostream>
```





Πρακτικές συμβουλές προγραμματισμού

1. Γράφουμε απλά και κατανοητά προγράμματά, ώστε ο (ίδιος ή άλλος) προγραμματιστής να μπορεί (πολύ) αργότερα να καταλάβει, τροποποιήσει ή/και επεκτείνει το πρόγραμμα.
2. Τα ονόματα σταθερών και μεταβλητών να υπονοούν τη χρήση τους.
3. Βάζουμε επεξηγηματικά σχόλια, ειδικά στα πιο δυσκολονόητα σημεία του προγράμματος.
4. Οι κενές γραμμών στα όρια των ενοτήτων του προγράμματος το κάνουν πιο ευανάγνωστο.
5. Η χρήση σταθερών διευκολύνει επόμενες αλλαγές αποτρέποντας ξεχασμένες παλιές τιμές είτε ακούσιες αλλαγές.
6. Θέτουμε αρχικές τιμές στις μεταβλητές του προγράμματος, ώστε η υλοποίηση να μην εξαρτάται αποκλειστικά από τυχόν αυτόματη αρχικοποίηση από το (κάθε) προγραμματιστικό περιβάλλον (μεταγλωττιστή).
7. Σπάμε μεγάλους υπολογισμούς σε απλούστερους για ευκολότερη κατανόηση του προγράμματος και αποφυγή λαθών.

Υπενθύμιση!





Η εντολή εκτύπωσης cout

(character output, βλ. https://www.stroustrup.com/bs_faq2.html#cout)

Γενική μορφή:

cout<<a;

cout<<a<<b<<c<<…<<x;

όπου **a, b, c, …, x** μπορεί να είναι σταθερές (αριθμητικές, χαρακτήρων ή συμβολοσειρές), μεταβλητές ή παραστάσεις της C++.





Παράδειγμα

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++. ”<<“It is
    really simple!”;
    return 0;
}
```

Αποτέλεσμα: Θα εκτυπωθεί το μήνυμα:

This is my first program in C++. It is really simple!





```
#include <iostream>
using namespace std;
int main( )
{
   cout <<“This is my first program in C++. ”;
   cout <<“It is really simple!”;
   return 0;
}
```

Αποτέλεσμα: Πάλι θα εκτυπωθεί το μήνυμα:

This is my first program in C++. It is really simple!

Πως θα τυπώσουμε τις δύο προτάσεις σε ξεχωριστές σειρές;

https://en.cppreference.com/cpp/language/bool_literal

```
#include <iostream>
int main()
{
        std::cout   << std::boolalpha
                    << true << '\n'
                    << false << '\n'
                    << std::noboolalpha
                    << true << '\n'
                    << false << '\n';
}
```

Input/output manipulators:
boolalpha, noboolalpha
https://en.cppreference.com/cpp/io/manip/boolalpha





https://en.cppreference.com/cpp/language/bool_literal

```
#include <iostream>
int main()
{
        std::cout       << std::boolalpha
                        << true << '\n'
                        << false << '\n'
                        << std::noboolalpha
                        << true << '\n'
                        << false << '\n';
}
```

Output:

```
true
false
1
0
```





Το σύμβολο **\n** και η λέξη **endl**

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.\nIt is
    really simple!”;
    return 0;
}
```

Αποτέλεσμα: Θα εκτυπωθεί το μήνυμα:

```
This is my first program in C++.
It is really simple!
```

Την ίδια εκτύπωση θα δώσει και η εκτέλεση των παρακάτω προγραμμάτων:





```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.”;
    cout <<“\nIt is really simple!”;
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main( )
{
   cout <<“This is my first program in C++.”
        <<‘\n’<<“It is really simple”;
   return 0;
}
```

Παρατήρηση: Η C++ αντιμετωπίζει το \n σαν **ένα** χαρακτήρα, ο οποίος ονομάζεται “newline character”





Η χρήση της λέξης endl

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.”<<endl;
    cout <<“It is really simple!”;
    return 0;
}
```

ή

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.”<<endl<<“It is really
    simple!”;
return 0;
}
```





Τι αποτέλεσμα θα είχε το πρόγραμμα;

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.”<<endl;
    cout <<“\nIt is really simple!”;
    return 0;
}
```

Αποτέλεσμα:

This is my first program in C++.

It is really simple!





Μπορούμε να έχουμε διευθετημένη εκτύπωση με χρήση του χειριστή εξόδου **setw(k),** όπου το **k** μπορεί να είναι ακέραιος θετικός, ή μεταβλητή ή παράσταση που παίρνουν τιμές ακέραιες θετικές. Το **k** καθορίζει το πλάτος (πλήθος χαρακτήρων) της επόμενης τιμής εξόδου και η τιμή αυτή θα είναι τοποθετημένη στις επόμενες **k** θέσεις στα δεξιά, π.χ.

```
#include <iostream>
#include <iomanip>  //header file απαραίτητο για το setw(k)
using namespace std;
int main ( )
{
    cout <<setw(14)<<"THESSALONIKI"<<setw(7)<<7<<endl;
    cout <<setw(14)<<"ATHENS"<<setw(7)<<8<<endl;
    cout <<setw(14)<<"LARISA"<<setw(7)<<9;
    return 0;
}
```

Αποτέλεσμα:

```
 THESSALONIKI      7
          ATHENS      8
           LARISA      9
```





Με το επόμενο πρόγραμμα «υπολογίζουμε» τις λύσεις της εξίσωσης δευτέρου βαθμού:

```
#include <iostream>
#include <math.h>   //header file απαραίτητο για τις
//μαθηματικές  συναρτήσεις. Προσοχή στην επέκταση h του
//ονόματος!
using namespace std;
int main ( )
{
    float a,b,c;
    float x1,x2;
    a=1;
    b=-5;
    c=6;
    x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
    x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
    cout <<“The solutions are:”<<“\nx1=”
                    <<x1<<“\nx2= ”<<x2;
    return 0;
}
```

```
#include <iostream>
#include <math.h>
using namespace std;
int main ( )
{
   float a=2;          //Είναι δυνατό να καταχωρούμε τιμές
   float b,c;          //σε μεταβλητές ταυτόχρονα με τη
                       //δήλωσή τους
    float x1,x2;
    b=-5;
    c=6;
    x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
    x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
    cout <<"The solutions are:"<<"\nx1="
                    <<x1<<"\nx2= "<<x2;
     return 0;
}
```





Με ποιο τρόπο θα κάνουμε το προηγούμενο πρόγραμμα *πιο γενικό και χρήσιμο*, ώστε ο χρήστης να δίνει τιμές στις a, b, c, και αυτό να επιστρέφει τις λύσεις της εξίσωσης;

Χρειαζόμαστε λοιπόν ένα τρόπο με τον οποίο να καθοδηγούμε το πρόγραμμα να ζητάει από το χρήστη τα δεδομένα.

Η εντολή **cin**

(character input, βλ.
https://www.geeksforgeeks.org/cpp/cin-in-c/ **)**

Γενική μορφή:

cin >>a;

cin >>a>>b>>…>>x;

όπου a, b, …, x είναι μεταβλητές.





```
#include <iostream>
#include <math.h>
using namespace std;
int main ( )
{
    float a,b,c;
    float x1,x2;
    cin >>a>>b>>c;      //δοκίμασε a=1, b=0, c= -1

    x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
    x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
    cout <<“The solutions are: ”<<“\nx1=”
                    <<x1<<“\nx2= ”<<x2;
     return 0;
}
```





Μια «πιο φιλική» προς το χρήστη μορφή του προηγούμενου προγράμματος:

```
#include <iostream>
#include <math.h>
using namespace std;
int main ( )
{
    float a,b,c;
    float x1,x2;
    cout <<“Assign values to a,b,c”;
    cin >>a>>b>>c;                          //δοκίμασε a=1, b=0, c= -1
    x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
    x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
    cout <<“The solutions are:”<<“\nx1=”
                    <<x1<<“\nx2= ”<<x2;
     return 0;
}
```





Μια «ακόμη πιο φιλική» προς το χρήστη μορφή του προηγούμενου προγράμματος:

```
#include <iostream>
#include <math.h>
using namespace std;
int main ( )
{
     float a,b,c;
     float x1,x2;
     cout <<“Assign value to a:”;
     cin >>a;
     cout <<“Assign value to b:”;
     cin >>b;
     cout <<“Assign value to c:”;
     cin >>c;
     x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
     x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
     cout <<“The solutions are:”<<“\nx1=”
                         <<x1<<“\nx2= ”<<x2;
      return 0;
}
```

Ασκήσεις W4.1

1) Θεωρούμε ότι ο κορμός ενός δένδρου είναι κυλινδρικός. Συνεπώς, ο όγκος του δένδρου V αν γνωρίζουμε την περίμετρό της βάσης του P, και το ύψος του H, θα δίνεται από τον τύπο :

$$V = \frac{P^2 \times H}{4 \times \pi}$$

όπου π=3.14.

Να γράψετε το κατάλληλο πρόγραμμα που θα υπολογίζει το V με δεδομένα τα H, P.





Ασκήσεις W4.2

2) Καταθέτουμε σε μια τράπεζα στην αρχή του κάθε έτους ένα σταθερό ποσό Α. Αν η κατάθεση γίνεται με επιτόκιο Ε%, τότε το κεφάλαιο Κ μετά την ν-οστή κατάθεση θα είναι :

$$K = \frac{A \times (1+E) \times \left[(1+E)^N - 1 \right]}{E}$$

Να γραφεί πρόγραμμα υπολογισμού του Κ αν είναι γνωστά τα Α, Ε και Ν.





Ασκήσεις W4.3

3) Έστω d_0, d_1 οι διάμετροι στη βάση και στο σημείο αποκορύφωσης ενός δένδρου. Αν το συνολικό μήκος του δένδρου είναι l, τότε ο όγκος V του δένδρου δίνεται από τον τύπο :

$$V = \frac{\pi}{4} \times \ell \times \left(\frac{d_0^2 + d_1^2}{2} \right)$$

Να γραφεί πρόγραμμα που να υπολογίζει τον όγκο του δένδρου, με δεδομένα το ύψος και τις δύο διαμέτρους.



