

# Εισαγωγή στη γλώσσα προγραμματισμού C++14

Σταμάτης Σταματιάδης  
stamatis@materials.uoc.gr

Τμήμα Επιστήμης και Τεχνολογίας Υλικών,  
Πανεπιστήμιο Κρήτης

## Σχετικά με το μάθημα

Διαλέξεις Τρίτη 17:00-19:00.

Ασκήσεις Δευτέρα 13:00-14:00.

Ιστοσελίδα Ιστοσελίδα Φυσικού → Εκπαίδευση → Ηλεκτρονικά Μαθήματα

- Βιβλιογραφία
- Στην ιστοσελίδα διατίθεται το βιβλίο που θα διδαχτεί.
  - Παρέχεται επιπλέον βιβλίο μέσω Εύδοξου:
    - “Προγραμματισμός με τη C++”, B. Stroustrup,
    - “Η ΓΛΩΣΣΑ C++ ΣΕ ΒΑΘΟΣ”, N. M. Χατζηγιαννάκης,
    - “C++: Από τη Θεωρία στην Εφαρμογή”, Γ. Τσελίκης.

Εξετάσεις Ασκήσεις συγγραφής κώδικα (με ανοιχτά βιβλία, σημειώσεις, κλπ.).

# Εξέλιξη της C++

## Δημιουργός

Bjarne Stroustrup  
(1979-1983) C With Classes

## Στόχος

Γρήγορη γλώσσα για systems programming  
που να υποστηρίζει object-oriented programming.

## Standards

- 1998 (C++98)
- 2003 (C++03)
- 2011 (C++11)
- 2014 (C++14)
- 2017 (C++17)
- 2020 (C++20)

## Προγραμματισμός στο Linux

1.
  - Δημιουργία κώδικα σε κειμενογράφο (emacs, gedit, vi, ...) ή IDE (Code::Blocks, eclipse, ...).
  - Αποθήκευση αρχείου με (προτεινόμενη) κατάληξη `.cpp`
2. Μεταγλώττιση (στο terminal ή στο IDE):  
`g++ -std=c++14 -Wall -Wextra -O2 arxio.cpp`
3. Εκτέλεση προγράμματος (στο terminal ή στο IDE):  
`./a.out`

## Παράδειγμα προγράμματος C++

```
#include <iostream>
/* Square of a real number */
int main()
{
    double a;
    std::cout << u8"Δώσε πραγματικό αριθμό: ";
    std::cin >> a;
    double b;
    b = a * a;
    std::cout << u8"Το τετράγωνο είναι: ";
    std::cout << b << '\n';
    return 0;
}
```

## Παράδειγμα (C) προς αποφυγή!

*This short program reads a spelt number (e.g. 'forty-two') and writes a corresponding decimal number (e.g. '42').*

```
long long n,u,m,b;main(e,r)char **r;{f\
or(;n++|| (e=getchar()|32)>=0;b="ynwtsflrabg"[n%=11]
-e?b:b*8+n)for(r=b%64-25;e<47&&b;b/=8)for(n=19;n;
n["1+DIY/.K430x9\
G(kC["]-42&255^b|| (m+=n>15?n:n>9?m%u*~-u:~(int)r?n+
!(int)r*16:n*16,b=0))u=111<<6177%n--*4;
printf("%llx\n",m);}
```

## Σχόλια

Δύο μορφές σχολίων:

- Κείμενο μεταξύ /\* και \*/, ανεξάρτητα από το πόσες γραμμές καταλαμβάνει.
- Κείμενο μεταξύ // και τέλους γραμμής.

## Είσοδος/Εξοδος δεδομένων

Αφού κάνουμε **#include** το `<iostream>`:

- Το 

```
std::cin >> a;
```

 διαβάζει τιμή στη μεταβλητή `a`.
- Το 

```
std::cout << a;
```

 τυπώνει την τιμή της ποσότητας `a`.
- Το 

```
std::cout << u8"Ελληνικό κείμενο";
```

 τυπώνει το κείμενο που γράφουμε εντός **διπλών** εισαγωγικών. Αν το κείμενο έχει ελληνικούς χαρακτήρες, βάζουμε **u8** πριν τα αρχικά εισαγωγικά.



## Τύποι ακέραιων ποσοτήτων (1/2)

- `short int`, `int`, `long int`, `long long int`,  
και οι αντίστοιχοι απρόσημοι τύποι
- `unsigned short int`, `unsigned int`, `unsigned long int`,  
`unsigned long long int`.

### Τιμές

Σειρά αριθμητικών ψηφίων χωρίς κενά, με πιθανό πρόσημο (+, -) και πιθανή κατάληξη:

`-1`, `5444`, `-45L`, `556UL`, `667ull`

## Τύποι ακέραιων ποσοτήτων (2/2)

### Παρατηρήσεις

- Εμπρόσημος ακέραιος τύπος για γενική χρήση: `int`.
- Συνήθης μέγιστη τιμή `int`:  $2147483647 (= 2^{31} - 1)$ .  
Συνήθης ελάχιστη τιμή `int`:  $-2147483648 (= -2^{31})$ .
- Απρόσημος ακέραιος τύπος για μετρητή ή δείκτη σε array:  
`std::size_t` από το `<stddef>`.
- Συνήθης μέγιστη τιμή `std::size_t`:  $18446744073709551615 (= 2^{64} - 1)$ .
- Τα ψηφία μπορούν να χωρίζονται με απόστροφο (`'`): `12'345'678`.
- Οι αριθμοί μπορούν να γραφούν εναλλακτικά στο δεκαεξαδικό (`0x`), στο οκταδικό (`0`) ή στο δυαδικό σύστημα (`0b`).

## Τύποι πραγματικών ποσοτήτων (1/2)

- **float** (απλής ακρίβειας, 6 σημαντικά ψηφία),
- **double** (διπλής ακρίβειας, 15 σημαντικά ψηφία),
- **long double** (εκτεταμένης ακρίβειας, 18 σημαντικά ψηφία).

### Τιμές

Σειρά αριθμητικών ψηφίων χωρίς κενά, με πιθανό πρόσημο (+, -), και

- **ή** τελεία (αντί για υποδιαστολή) που χωρίζει το ακέραιο από το δεκαδικό μέρος: Π.χ. 12.345, -1.02.
- **ή** το χαρακτήρα e (ή E) που χωρίζει τον αριθμό από τη δύναμη του 10 με την οποία πολλαπλασιάζεται. Π.χ. 123E2 ( $\equiv 123 \times 10^2 \equiv 12300.0$ ), -12e-1 ( $\equiv -1.2$ ),
- **ή** συνδυασμό των παραπάνω: Π.χ. -1.2E-2 ( $\equiv -0.012$ ),

και πιθανή κατάληξη: Π.χ. 3E4f, 3E4L.

## Τύποι πραγματικών ποσοτήτων (2/2)

### Παρατηρήσεις

- Ποιον τύπο πρέπει να χρησιμοποιούμε; **double**
- Συνήθης μέγιστη απόλυτη τιμή **double**:  $\approx 10^{308}$ .
- Συνήθης ελάχιστη απόλυτη τιμή **double**:  $\approx 10^{-308}$ .
- Πώς γράφουμε σε **double** το  $10^{-6}$ ; **1E-6**

## Αριθμητικοί τελεστές μεταξύ ακέραιων ποσοτήτων

Άθροισμα	+	Π.χ.	$5 + 7$
Διαφορά	-	Π.χ.	$5 - 7$
Γινόμενο	*	Π.χ.	$2 * 3$
Πηλίκο	/	Π.χ.	$7/2 \rightarrow 3$ (πόσες φορές «χωρά» το 2 στο 7).
Υπόλοιπο	%	Π.χ.	$7\%2 \rightarrow 1$

Γενικά ισχύει (για ακέραια  $a, b$  με  $b > 0$ )

- $0 \leq a\%b < b$ ,
- $a = (a/b) * b + a\%b$ .

### Ύψωση σε δύναμη

Με συνάρτηση: `std::pow(a,b)`  $\rightarrow a^b$  από το `<cmath>`.

Τα  $a, b$  είναι ακέραια.

Το αποτέλεσμα είναι **double** (γιατί οι τιμές τους μετατρέπονται σε **double** κατά την κλήση).

## Αριθμητικοί τελεστές μεταξύ πραγματικών ποσοτήτων

Άθροισμα	+	Π.χ.	$5.0 + 7.0$
Διαφορά	-	Π.χ.	$5.0f - 7.0f$
Γινόμενο	*	Π.χ.	$2.0 * 3e2$
Λόγος	/	Π.χ.	$7.0/2.0$

### Ύψωση σε δύναμη

Με συνάρτηση: `std::pow(a,b)` →  $a^b$  από το `<cmath>`.

Τα  $a, b$  είναι πραγματικοί και το αποτέλεσμα έχει τον ίδιο τύπο με αυτούς.

## Γενικές παρατηρήσεις στους αριθμητικούς τελεστές

### Κανόνες

- Τελεστές που δρουν μεταξύ ποσοτήτων **ίδιου τύπου** δίνουν αποτέλεσμα **αυτού του τύπου**.
- Τελεστές που δρουν μεταξύ ποσοτήτων **διαφορετικού τύπου** προκαλούν μετατροπή της **τιμής** της ποσότητας «χαμηλότερης» ακρίβειας στον τύπο με την «υψηλότερη» ακρίβεια πριν από την πράξη.

**Παράδειγμα:** `int` σε `double`:  $2*3.0 \rightarrow 2.0*3.0 \rightarrow 6.0$ .

**Εξαίρεση:** Ποσότητες τύπου `short int` γίνονται `int` πριν την πράξη.

## Σύγκριση πραγματικών αριθμών

Τι θα τυπωθεί με τον παρακάτω κώδικα;

```
std::cout << 0.1+0.2-0.3 << '\n';  
std::cout << 0.1-0.3+0.2 << '\n';
```

### Απάντηση

5.55112e-17

2.77556e-17

Όχι 0, ούτε καν ο ίδιος αριθμός.

### Κανόνας

Οι πράξεις μεταξύ πραγματικών αριθμών δεν είναι απόλυτα ακριβείς. Επομένως, **δεν συγκρίνουμε ποτέ για ισότητα πραγματικές ποσότητες που τουλάχιστον η μία προέκυψε από πράξεις.**



## Συντμήσεις αριθμητικών τελεστών

$x = x + a;$  →  $x += a;$

$x = x * a;$  →  $x *= a;$

$x = x \% a;$  →  $x %= a;$

$x = x - a;$  →  $x -= a;$

$x = x / a;$  →  $x /= a;$

### Ειδικές περιπτώσεις

$x = x + 1;$  →  $x += 1;$  →  $++x;$  ή  $x++;$

$x = x - 1;$  →  $x -= 1;$  →  $--x;$  ή  $x--;$

### Παρατηρήσεις

- Οι συντμήσεις μπορεί να εμφανιστούν σε σύνθετο κώδικα, π.χ.  
 $b = a += 5;$  →  $a = a + 5;$   $b = a;$
- Οι σύνθετες εντολές  $b = ++a;$  και  $b = a++;$  **διαφέρουν:**  
 $b = ++a;$  →  $a = a + 1;$   $b = a;$   
 $b = a++;$  →  $b = a;$   $a = a + 1;$

## Σχετικές προτεραιότητες αριθμητικών τελεστών

Πολύ Υψηλή Παρενθέσεις ( )

Υψηλή ++, --

Μεσαία \*, /, %

Χαμηλή +, -

Πολύ Χαμηλή +=, -=, \*=, /=, %=

### Παρατήρηση

Τελεστές με ίδια προτεραιότητα σε μια έκφραση εκτελούνται από αριστερά προς τα δεξιά.

## Μεταβλητή (1/3)

### Κανόνας

Κάθε μεταβλητή προτού χρησιμοποιηθεί **πρέπει να δηλωθεί**, κατά προτίμηση λίγο πριν χρησιμοποιηθεί. Αν γνωρίζουμε την αρχική της τιμή, καλό είναι να κάνουμε δήλωση με απόδοση αρχικής τιμής.

### Δήλωση

**τύπος** όνομα\_μεταβλητής;

### Παραδείγματα

```
int a;  
double b;
```

Είναι πιο ευανάγνωστος ο κώδικας αν δηλώνουμε κάθε ποσότητα σε ξεχωριστή γραμμή.

## Μεταβλητή (2/3)

Δήλωση με απόδοση αρχικής τιμής

*τύπος όνομα\_μεταβλητής{αρχική τιμή};*

Παραδείγματα

```
int a{3};  
double b{4.5};
```

Παρατηρήσεις

- **OXI** `int a{4.3};`
- Επιτρέπεται το `int a = 4.3; ;`
- Κενά άγκιστρα → default τιμή (για αριθμούς, 0).
- Αν δεν δώσουμε αρχική τιμή, οι μεταβλητές των ενσωματωμένων τύπων αποκτούν **απροσδιόριστη τιμή**.

## Μεταβλητή (3/3)

Αυτόματη αναγνώριση τύπου από την αρχική τιμή  
***auto*** όνομα\_μεταβλητής = αρχική τιμή;

### Παραδείγματα

```
auto a = 3;
```

```
auto b = 3.6;
```

Το a είναι **int** με τιμή 3, το b **double** με τιμή 3.6.

### Παρατήρηση

```
OXI auto a{4.3};
```

(δεν κάνει αυτό που νομίζουμε.)

# Ασκήσεις

Πρώτο κεφάλαιο

1,2,3

Δεύτερο κεφάλαιο

1,2,3