

Εισαγωγή στη γλώσσα προγραμματισμού C++14

Σταμάτης Σταματιάδης
stamatis@materials.uoc.gr

Τμήμα Επιστήμης και Τεχνολογίας Υλικών,
Πανεπιστήμιο Κρήτης

Σχετικά με το μάθημα

Διαλέξεις–Ασκήσεις Παρασκευή 17:00-20:00 στην αίθουσα υπολογιστών 2.

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, Ν. Μ. Χατζηγιαννάκης

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)

Προγραμματισμός στο Linux

1.
 - Δημιουργία κώδικα σε κειμενογράφο (emacs, gedit, vi, ...) ή IDE (Code::Blocks, eclipse, ...).
 - Αποθήκευση αρχείου με (προτεινόμενη) κατάληξη `.cpp`

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 arxeio.cpp`

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`

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 arxheio.cpp`
3. Εκτέλεση προγράμματος (στο terminal ή στο IDE):
`./a.out`

Παρατήρηση

Στους υπολογιστές του εργαστηρίου ο compiler είναι ο `g++-6.4.0` .

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 προς αποφυγή!

```
#include <stdio.h>
int l;int main(int o,char **O,
int I){char c,*D=O[1];if(o>0){
for(l=0;D[l]
];D[l
++]-=10){D [l++]-=120;D[l]-=
110;while (!main(0,0,l))D[l]
+= 20; putchar((D[l]+1032)
/20 ) ;}putchar(10);}else{
c=o+ (D[I]+82)%10-(I>1/2)*
(D[I-1+I]+72)/10-9;D[I]+=I<0?0
:!(o=main(c/10,0,I-1))*((c+999
)%10-(D[I]+92)%10);}return o;}
```

Τύποι ακέραιων ποσοτήτων

`short int`, `int`, `long int`, `long long int`,
με `signed` και `unsigned` παραλλαγές.

Τιμές

-1, 5444, 45L, 556UL, 667ULL (δεκαδικό σύστημα).

Τύποι ακέραιων ποσοτήτων

`short int`, `int`, `long int`, `long long int`,
με `signed` και `unsigned` παραλλαγές.

Τιμές

-1, 5444, 45L, 556UL, 667ULL (δεκαδικό σύστημα).

Παρατηρήσεις

- Δεν υπάρχει κατάληξη για `short int`.
- Οι αριθμοί μπορούν να γραφούν στο δεκαεξαδικό (0x), στο οκταδικό (0) ή στο δυαδικό σύστημα (0b).
- Τα ψηφία μπορούν να χωρίζονται με απόστροφο (').

Τύποι ακέραιων ποσοτήτων

`short int`, `int`, `long int`, `long long int`,
με `signed` και `unsigned` παραλλαγές.

Τιμές

-1, 5444, 45L, 556UL, 667ULL (δεκαδικό σύστημα).

Παρατηρήσεις

- Δεν υπάρχει κατάληξη για `short int`.
- Οι αριθμοί μπορούν να γραφούν στο δεκαεξαδικό (0x), στο οκταδικό (0) ή στο δυαδικό σύστημα (0b).
- Τα ψηφία μπορούν να χωρίζονται με απόστροφο (').

Ακέραιος τύπος για δείκτης σε array

`std::size_t` από το `<cstdlib>`.

Τύποι πραγματικών ποσοτήτων

`float` (απλής ακρίβειας, 6 σημαντικά ψηφία).
Π.χ. `-3.2f`, `3E4F`, `3.1E-3F`.

Τύποι πραγματικών ποσοτήτων

float (απλής ακρίβειας, 6 σημαντικά ψηφία).

Π.χ. `-3.2f`, `3E4F`, `3.1E-3F`.

double (διπλής ακρίβειας, 15 σημαντικά ψηφία).

Π.χ. `-3.2`, `3E4`, `3.1E-3`.

Τύποι πραγματικών ποσοτήτων

float (απλής ακρίβειας, 6 σημαντικά ψηφία).

Π.χ. `-3.2f`, `3E4F`, `3.1E-3F`.

double (διπλής ακρίβειας, 15 σημαντικά ψηφία).

Π.χ. `-3.2`, `3E4`, `3.1E-3`.

long double (εκτεταμένης ακρίβειας, 18 σημαντικά ψηφία).

Π.χ. `-3.2L`, `3E4L`, `3.1E-3L`.

Τύποι πραγματικών ποσοτήτων

float (απλής ακρίβειας, 6 σημαντικά ψηφία).

Π.χ. `-3.2f`, `3E4F`, `3.1E-3F`.

double (διπλής ακρίβειας, 15 σημαντικά ψηφία).

Π.χ. `-3.2`, `3E4`, `3.1E-3`.

long double (εκτεταμένης ακρίβειας, 18 σημαντικά ψηφία).

Π.χ. `-3.2L`, `3E4L`, `3.1E-3L`.

Παρατηρήσεις

- Ποιον τύπο πρέπει να χρησιμοποιούμε;

Τύποι πραγματικών ποσοτήτων

float (απλής ακρίβειας, 6 σημαντικά ψηφία).

Π.χ. `-3.2f`, `3E4F`, `3.1E-3F`.

double (διπλής ακρίβειας, 15 σημαντικά ψηφία).

Π.χ. `-3.2`, `3E4`, `3.1E-3`.

long double (εκτεταμένης ακρίβειας, 18 σημαντικά ψηφία).

Π.χ. `-3.2L`, `3E4L`, `3.1E-3L`.

Παρατηρήσεις

- Ποιον τύπο πρέπει να χρησιμοποιούμε;
- Πώς γράφουμε το 10^{-6} ;

Αριθμητικοί Τελεστές (1/5)

Μεταξύ ακέραιων ποσοτήτων

Άθροισμα + Π.χ. 5 + 7

Διαφορά - Π.χ. 5 - 7

Γινόμενο * Π.χ. 2 * 3

Αριθμητικοί Τελεστές (1/5)

Μεταξύ ακέραιων ποσοτήτων

Άθροισμα	+	Π.χ. $5 + 7$
Διαφορά	-	Π.χ. $5 - 7$
Γινόμενο	*	Π.χ. $2 * 3$
Πηλίκο	/	Π.χ. $7/2 \rightarrow 3$

Αριθμητικοί Τελεστές (1/5)

Μεταξύ ακέραιων ποσοτήτων

Άθροισμα	+	Π.χ. $5 + 7$
Διαφορά	-	Π.χ. $5 - 7$
Γινόμενο	*	Π.χ. $2 * 3$
Πηλίκο	/	Π.χ. $7/2 \rightarrow 3$
Υπόλοιπο	%	Π.χ. $7\%2 \rightarrow 1$

Αριθμητικοί Τελεστές (2/5)

Μεταξύ πραγματικών ποσοτήτων

Άθροισμα	+	Π.χ. $5.0 + 7.0$
Διαφορά	-	Π.χ. $5.0f - 7.3f$
Γινόμενο	*	Π.χ. $2.0 * 3e0$

Αριθμητικοί Τελεστές (2/5)

Μεταξύ πραγματικών ποσοτήτων

Άθροισμα	+	Π.χ. $5.0 + 7.0$
Διαφορά	-	Π.χ. $5.0f - 7.3f$
Γινόμενο	*	Π.χ. $2.0 * 3e0$
Λόγος	/	Π.χ. $3e0/2e0 \rightarrow 1.5$

Αριθμητικοί Τελεστές (3/5)

Κανόνας

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. Π.χ. **int** σε **double**.

Εξαιρέση: **short int** (γίνεται **int** πριν τη πράξη).

Αριθμητικοί Τελεστές (3/5)

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. Π.χ. **int** σε **double**.

Εξαιρέση: **short int** (γίνεται **int** πριν τη πράξη).

Παρατηρήσεις

- Ύψωση σε δύναμη;

Αριθμητικοί Τελεστές (3/5)

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. Π.χ. **int** σε **double**.

Εξαιρέση: **short int** (γίνεται **int** πριν τη πράξη).

Παρατηρήσεις

- Ύψωση σε δύναμη;
`std::pow(a, b)` → a^b από το `<cmath>`.

Αριθμητικοί Τελεστές (3/5)

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. Π.χ. **int** σε **double**.

Εξαιρέση: **short int** (γίνεται **int** πριν τη πράξη).

Παρατηρήσεις

- Ύψωση σε δύναμη;
`std::pow(a, b) → a^b` από το `<cmath>`.
- Πώς ΔΕΝ γράφουμε το 10^{-6} ;

Αριθμητικοί Τελεστές (4/5)

Συντμήσεις

`x = x + a;` → `x += a;`

Αριθμητικοί Τελεστές (4/5)

Συντμήσεις

$x = x + a;$ → $x += a;$

$x = x * a;$ → $x *= a;$

$x = x \% a;$ → $x \% = a;$

$x = x - a;$ → $x -= a;$

$x = x / a;$ → $x /= a;$

Αριθμητικοί Τελεστές (4/5)

Συντμήσεις

$x = x + a;$ → $x += a;$

$x = x * a;$ → $x *= a;$

$x = x \% a;$ → $x %= a;$

$x = x - a;$ → $x -= a;$

$x = x / a;$ → $x /= a;$

Ειδικές περιπτώσεις

$x = x + 1;$ → $x += 1;$ → $++x;$

Αριθμητικοί Τελεστές (4/5)

Συντμήσεις

$x = x + a;$ → $x += a;$

$x = x * a;$ → $x *= a;$

$x = x \% a;$ → $x %= a;$

$x = x - a;$ → $x -= a;$

$x = x / a;$ → $x /= a;$

Ειδικές περιπτώσεις

$x = x + 1;$ → $x += 1;$ → $++x;$

$x = x - 1;$ → $x -= 1;$ → $--x;$

Αριθμητικοί Τελεστές (5/5)

Σχετικές Προτεραιότητες

Πολύ Υψηλή Παρενθέσεις ()

Υψηλή ++, --

Μεσαία *, /, %

Χαμηλή +, -

Πολύ Χαμηλή +=, -=, *=, /=, %=

Μεταβλητή (1/4)

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.

Μεταβλητή (1/4)

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;

Μεταβλητή (1/4)

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;
```

Μεταβλητή (1/4)

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 (2/4)

Κανόνες σχηματισμού ονόματος

- Επιτρεπτοί χαρακτήρες: a-z, A-Z, 0-9, και _.
- Μήκος: οποιοδήποτε.
- Δεν επιτρέπεται να αρχίζει από αριθμητικό ψηφίο.
- Δεν επιτρέπεται να αποτελεί ενσωματωμένη λέξη της C++.
- Κεφαλαία και πεζά γράμματα **είναι διαφορετικά**.

Μεταβλητή (3/4)

Δήλωση με απόδοση αρχικής τιμής

τύπος όνομα_μεταβλητής{αρχική τιμή};

Μεταβλητή (3/4)

Δήλωση με απόδοση αρχικής τιμής

τύπος όνομα_μεταβλητής{αρχική τιμή};

Παραδείγματα

```
int a{3};
```

```
double b{4.5};
```

Μεταβλητή (3/4)

Δήλωση με απόδοση αρχικής τιμής

τύπος όνομα_μεταβλητής{αρχική τιμή};

Παραδείγματα

```
int a{3};
```

```
double b{4.5};
```

Παρατηρήσεις

- **OXI** `int a{4.3};`

Μεταβλητή (3/4)

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; ;`

Μεταβλητή (3/4)

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).

Μεταβλητή (4/4)

Αυτόματη αναγνώριση τύπου από την αρχική τιμή

auto όνομα_μεταβλητής = αρχική τιμή;

Μεταβλητή (4/4)

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.

Μεταβλητή (4/4)

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.)

Εντολή εκχώρησης τιμής

μεταβλητή = [γενική έκφραση] ;

Πρώτα εκτελούνται όλες οι πράξεις, κλήσεις συναρτήσεων κλπ. που εμφανίζονται στο δεξί μέλος.

Εντολή εκχώρησης τιμής

μεταβλητή = [γενική έκφραση] ;

Πρώτα εκτελούνται όλες οι πράξεις, κλήσεις συναρτήσεων κλπ. που εμφανίζονται στο δεξί μέλος.

Κατόπιν, το αποτέλεσμα **μετατρέπεται** (αν χρειάζεται) στον τύπο της **(υποχρεωτικά) μεταβλητής** του αριστερού μέλους και η τιμή που προκύπτει εκχωρείται σε αυτή.

Εντολή εκχώρησης τιμής

μεταβλητή = [γενική έκφραση] ;

Πρώτα εκτελούνται όλες οι πράξεις, κλήσεις συναρτήσεων κλπ. που εμφανίζονται στο δεξί μέλος.

Κατόπιν, το αποτέλεσμα **μετατρέπεται** (αν χρειάζεται) στον τύπο της **(υποχρεωτικά) μεταβλητής** του αριστερού μέλους και η τιμή που προκύπτει εκχωρείται σε αυτή.

Παραδείγματα

```
int a;
```

```
float b;
```

```
a = 2.5 * 3; // a ← 7
```

```
b = 1.00000001; // b ← 1.0f
```