

# ΥΠΟΛΟΓΙΣΤΕΣ ΙΙ

## ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ΠΡΑΞΕΙΣ ΜΕΤΑΒΛΗΤΕΣ

1

## Τύποι δεδομένων

Η γλώσσα προγραμματισμού C++ υποστηρίζει τους παρακάτω τύπους δεδομένων:

- 1) Ακέραιοι αριθμοί (*int*).
- 2) Πραγματικοί αριθμοί διπλής ακρίβειας (*double*).
- 3) Πραγματικοί αριθμοί απλής ακρίβειας (*float*).
- 4) Χαρακτήρες (*char*).

2

## Ακέραιοι αριθμοί (*int*)

- Πρόκειται για αριθμούς χωρίς δεκαδικά ψηφία.
- Για την αναπαράσταση των ακέραιων αριθμών χρησιμοποιούνται 4 bytes μνήμης.
- Λαμβάνουν τιμές στην περιοχή -2147483648 έως 2147483647.

3

## Πράξεις μεταξύ ακεραίων αριθμών

Μεταξύ δύο ακεραίων αριθμών μπορούν να γίνουν οι παρακάτω πράξεις χρησιμοποιώντας τον αντίστοιχο τελεστή:

- + Πρόσθεση
- Αφαίρεση
- \* Πολλαπλασιασμός
- / Διάρηση
- % Υπόλοιπο διαίρεσης

Το αποτέλεσμα μιας πράξης μεταξύ ακεραίων αριθμών είναι **πάντα** ακέραιος αριθμός.

**Προσοχή:** Δεν υπάρχει τελεστής για ύψωση σε δύναμη.

4

## Ακέραια διαίρεση

Η διαίρεση δύο ακεραίων αριθμών δίνει ως αποτέλεσμα επίσης ένα ακέραιο αριθμό.

Κατά τη διαίρεση δύο ακεραίων αριθμών τυχόν δεκαδικά ψηφία που προκύπτουν κατά τη διαίρεση **αποκόπτονται**.

Παραδείγματα:

$9/2$       Αποτέλεσμα: 4 (όχι 4.5)  
 $1/3$       Αποτέλεσμα: 0 (όχι 0.33333...)  
 $2/3$       Αποτέλεσμα: 0 (όχι 0.66666...)  
 $-6/4$      Αποτέλεσμα: -1 (όχι -1.5)

Η λειτουργία αυτή ονομάζεται **ακέραια διαίρεση**.

## Παράδειγμα #1

$3+5$	Αποτέλεσμα: 8
$4-8$	Αποτέλεσμα: -4
$6*2$	Αποτέλεσμα: 12
$8/3$	Αποτέλεσμα: 2
$7\%3$	Αποτέλεσμα: 1
$12\%5$	Αποτέλεσμα: 2

## Σύνθετες αριθμητικές παραστάσεις

Για να υπολογιστούν πιο σύνθετες αριθμητικές παραστάσεις όπως πχ.

$$2+4*3/2-7$$

$$6+(4/2-8)\%2$$

έχει ανατεθεί σε κάθε τελεστή **προτεραιότητα** και **προσεταιριστικότητα**.

<u>Τελεστής</u>	<u>Προτεραιότητα</u>	<u>Προσεταιριστικότητα</u>
* / %	Υψηλή	Από αριστερά προς δεξιά
+ -	Χαμηλή	Από αριστερά προς δεξιά

## Κανόνες υπολογισμού σύνθετων παραστάσεων

Σε μια σύνθετη αριθμητική παράσταση:

1. Πρώτα γίνονται οι πράξεις με τη μεγαλύτερη προτεραιότητα.
2. Μεταξύ πράξεων με την ίδια προτεραιότητα η σειρά των πράξεων καθορίζεται από την προσεταιριστικότητα.
3. Εάν υπάρχουν παρενθέσεις, τότε πρώτα γίνονται οι πράξεις εντός του πιο εσωτερικού ζεύγους παρενθέσεων.

## Πραγματικοί αριθμοί διπλής ακρίβειας (double)

- Πρόκειται για αριθμούς με δεκαδικά ψηφία
- Απαιτούνται 8 bytes για την αποθήκευσή τους.
- Έχουν περίπου 15 σημαντικά ψηφία.
- Λαμβάνουν τιμές στην περιοχή  $-10^{308} \dots -10^{-308}$   
 $10^{-308} \dots 10^{308}$
- Στη C++ οποιαδήποτε αριθμητική σταθερά με δεκαδικά ψηφία θεωρείται πραγματικός αριθμός διπλής ακρίβειας.

9

## Πραγματικοί αριθμοί: Επιστημονική αναπαράσταση

Αριθμός	Επιστ. αναπαράσταση	Σημαίνει
32.76	3.276e1	$3.276 \times 10^1$
-98541.34	-9.854134e4	$-9.854134 \times 10^4$
0.000035	3.5e-5	$3.5 \times 10^{-5}$

10

## Πραγματικοί αριθμοί απλής ακρίβειας (float)

- Απαιτούνται 4 bytes για την αποθήκευσή τους.
- Έχουν περίπου 7 σημαντικά ψηφία.
- Λαμβάνουν τιμές στην περιοχή  $-10^{38} \dots -10^{-38}$   
 $10^{-38} \dots 10^{38}$
- Για να ξεχωρίσουν από τους πραγματικούς διπλής ακρίβειας γράφονται με το επίθεμα f

11

## Παράδειγμα #2

3.14	Διπλής ακρίβειας (double)
-1.2e4	Διπλής ακρίβειας (double)
67.51f	Απλής ακρίβειας (float)
2.2e-3f	Απλής ακρίβειας (float)

12

## Πράξεις μεταξύ πραγματικών αριθμών

Μεταξύ πραγματικών αριθμών γίνονται οι πράξεις:

+ - \* /

Η διαίρεση γίνεται κατά το συνήθη τρόπο, δηλαδή αν προκύψουν δεκαδικά ψηφία, αυτά παραμένουν στον αριθμό.

Σημείωση: Κάποιοι μεταφραστές επιτρέπουν και το υπόλοιπο διαίρεσης % ως πράξη, χωρίς αυτό όμως να περιλαμβάνεται στο πρότυπο της γλώσσας.

13

## Μικτή αριθμητική

Όταν σε μια πράξη συμμετέχουν αριθμοί από δύο διαφορετικούς τύπους δεδομένων, τότε:

τα δεδομένα **προάγονται** αυτόματα στον ανώτερο από τους δύο τύπους

και κατά συνέπεια το αποτέλεσμα θα είναι του ανώτερου τύπου.

Οι τύποι δεδομένων ιεραρχούνται ως εξής:

double	Ανώτερος
float	
int	Κατώτερος

14

## Συναρτήσεις

Η γλώσσα προγραμματισμού C++ περιέχει μια σειρά από ενσωματωμένες μαθηματικές συναρτήσεις.

Για να χρησιμοποιήσουμε οποιαδήποτε μαθηματική συνάρτηση πρέπει να προσθέσουμε στην αρχή του προγράμματος:

```
#include <cmath>
```

15

## Συναρτήσεις

Για να υπολογίσουμε

Γράφουμε

$\sqrt{x}$	Τετραγωνική ρίζα	<code>sqrt(x)</code>
$e^x$	Εκθετικό	<code>exp(x)</code>
$ x $	Απόλυτη τιμή	<code>abs(x)</code>
$\ln x$	Νεπέρειος λογάριθμος	<code>log(x)</code>
$\log x$	Λογάριθμος με βάση 10	<code>log10(x)</code>
$\sin x$	Ημίτονο	<code>sin(x)</code>
$\cos x$	Συνημίτονο	<code>cos(x)</code>
$\tan x$	Εφαπτομένη	<code>tan(x)</code>
$\sin^{-1}x$	Τόξο ημιτόνου	<code>asin(x)</code>
$\cos^{-1}x$	Τόξο συνημιτόνου	<code>acos(x)</code>
$\tan^{-1}x$	Τόξο εφαπτομένης	<code>atan(x)</code>

16

## Συναρτήσεις

Όλες οι συναρτήσεις δέχονται ως όρισμα αριθμούς `float` ή `double` και επιστρέφουν αποτέλεσμα του αντίστοιχου τύπου. Πχ.

`sin(1.0)`      Σωστό  
`sin(1)`        Λάθος

Οι τριγωνομετρικές συναρτήσεις δέχονται το όρισμα σε **ακτίνια**, όχι μοίρες.

Οι αντίστροφες τριγωνομετρικές συναρτήσεις επιστρέφουν αποτέλεσμα σε **ακτίνια**, όχι μοίρες.

17

## Οι αριθμοί π, e

Για να υπολογίσουμε τον αριθμό π γράφουμε:

`acos(-1.0)`

Για να υπολογίσουμε τον αριθμό e (βάση των φυσικών λογαρίθμων) γράφουμε:

`exp(1.0)`

18

## Ύψωση σε δύναμη

Η ύψωση σε δύναμη  $a^b$  γίνεται με τη συνάρτηση `pow(a,b)`

Η βάση (a) και ο εκθέτης (b) μπορεί να είναι:

<u>Τύπος βάσης</u>	<u>Τύπος εκθέτη</u>	<u>Τύπος αποτελέσματος</u>
float	float	float
double	double	double
float	int	float
double	int	double

**Προσοχή:** Η βάση δεν μπορεί να είναι ακέραιος αριθμός.  
Πχ. `pow(4,2)` είναι ΛΑΘΟΣ

19

## Παράδειγμα #3

Όγκος σφαίρας με ακτίνα r:  $\frac{4}{3}\pi r^3$

`4*acos(-1.0)*pow(r,3)/3`

Απόσταση μεταξύ των σημείων  $(x_1, y_1)$  και  $(x_2, y_2)$ :

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

`sqrt(pow(x1-x2,2)+pow(y1-y2,2))`

20

## Χαρακτήρες (char)

- Καταλαμβάνουν 1 byte μνήμης.
- Υπάρχουν 256 διαφορετικοί χαρακτήρες.
- Ορισμένοι χαρακτήρες δεν εμφανίζονται στην οθόνη (πχ. esc, backspace)
- Κάθε χαρακτήρας αντιστοιχεί σε ένα ακέραιο αριθμό.
- Οι χαρακτήρες **a..z**, **A..Z** και **0..9** βρίσκονται στη σειρά, δηλαδή αντιστοιχούν σε διαδοχικούς ακέραιους αριθμούς.
- Γράφονται μέσα σε απλές αποστρόφους. Πχ. 'a' '8' 'E'
- Ορισμένοι ειδικοί χαρακτήρες γράφονται χρησιμοποιώντας το σύμβολο \ και ονομάζονται χαρακτήρες διαφυγής. Πχ. '\n'

21

## Χαρακτήρες διαφυγής

### Χαρακτήρας Τι σημαίνει

\n	Αλλαγή γραμμής (new line)
\b	Μία θέση πίσω (backspace)
\f	Αλλαγή σελίδας (form feed)
\r	Αρχή γραμμής (carriage return)
\t	Στηλοθέτης (tab)
\\	Ο χαρακτήρας \
\'	Απλή απόστροφος
\"	Διπλή απόστροφος
\0	Χαρακτήρας που αντιστοιχεί στον ακέραιο 0

22

## Μεταβλητές

Οι μεταβλητές είναι συμβολικά ονόματα που δίνουμε σε θέσεις μνήμης όπου αποθηκεύονται αριθμοί (ή άλλοι τύποι δεδομένων).

Κάθε μεταβλητή έχει **όνομα** και **τύπο**, που δηλώνονται υποχρεωτικά στην αρχή του προγράμματος.

23

## Κανόνες ονομάτων

- Επιτρέπονται μόνο λατινικοί χαρακτήρες, αριθμοί και το σύμβολο \_
- Ο πρώτος χαρακτήρας πρέπει να είναι γράμμα.
- Κεφαλαία και μικρά θεωρούνται διαφορετικά.
- Δεν υπάρχει μέγιστο μήκος στο όνομα, όμως πρακτικά κάθε μεταφραστής επιβάλλει ένα μέγιστο μήκος (πχ 32 χαρακτήρες).

Οι ίδιοι κανόνες ισχύουν και για άλλα ονόματα, πχ. ονόματα υποπρογραμμάτων.

24

## Δηλώσεις μεταβλητών

Οι μεταβλητές δηλώνονται στην αρχή κάθε προγράμματος:

```
#include <iostream>
using namespace std;

int main ( )
{
    int k, m;
    float a;
    double x, y, z;

    ... εντολές ...
}
```

25

## Ανάθεση τιμών

Για να δώσουμε τιμή σε μια μεταβλητή χρησιμοποιούμε τον **τελεστή** ανάθεσης τιμής =

*Μεταβλητή = Αριθμητική παράσταση*

**Αριστερά** του = είναι πάντα μια **μεταβλητή**  
**Δεξιά** του = είναι πάντα μια **αριθμητική παράσταση**

Παραδείγματα:

```
r = 5.
d = b*b-4*a*c
area = acos(-1.0)*r*r
x1 = (-b+sqrt(d))/(2*a)
```

26

## Ανάθεση τιμών στη δήλωση

Μπορούμε να δώσουμε αρχική τιμή σε μια μεταβλητή μέσω της δήλωσής της. Πχ.

```
#include <iostream>
using namespace std;

int main ( )
{
    int k=0;
    double x=1.45;

    ... εντολές ...
}
```

Στα δεξιά του = πρέπει να είναι μια σταθερή τιμή.

27

## Τελεστές αύξησης και μείωσης

Για την ειδική περίπτωση της πράξης μεταξύ ακεραίων:

`k = k+1;`

η C++ έχει ένα ειδικό τελεστή. Μπορούμε να γράψουμε:

`++k;` (προθεματική μορφή)

ή

`k++;` (επιθεματική μορφή)

28

## Τελεστές αύξησης και μείωσης

Στην προθεματική μορφή:

Πρώτα αυξάνεται η τιμή της μεταβλητής και κατόπιν η τιμή χρησιμοποιείται.

Στην επιθεματική μορφή:

Πρώτα χρησιμοποιείται η τιμή της μεταβλητής και κατόπιν η τιμή της μεταβλητής αυξάνεται.

Παράδειγμα:

```
k = 5;
a = ++k;
```

Η k γίνεται 6  
Η a γίνεται 6

```
k = 5;
a = k++;
```

Η k γίνεται 6  
Η a γίνεται 5

29

## Τελεστές αύξησης και μείωσης

- Αντίστοιχα με τον τελεστή αύξησης ++ υπάρχει ο τελεστής μείωσης -- σε προθεματική και επιθεματική μορφή.
- Οι τελεστές ++ και -- χρησιμοποιούνται μόνο σε ακέραιους αριθμούς.
- Εντός μιας αριθμητικής παράστασης οι τελεστές ++ και -- έχουν υψηλότερη προτεραιότητα από τους τελεστές + - \* / %
- Η προσηταιριστικότητα των τελεστών ++ και -- είναι από δεξιά προς τα αριστερά.

30

## Παράδειγμα #4

Ποια είναι η τιμή των μεταβλητών n, j μετά από τις εντολές:

```
n = 10;
j = ++n % 4;
```

Η n γίνεται 11  
Η j γίνεται 3

31

## Παράδειγμα #5

Ποια είναι η τιμή των μεταβλητών m, k μετά από τις εντολές:

```
m = 4;
k = 2+ m++;
```

Η m γίνεται 5  
Η k γίνεται 6

32



## Άλλοι τελεστές ανάθεσης τιμής

Η ανάθεση τιμής:  $k = k+m$

μπορεί να γραφεί στη C++ ως:  $k += m$

Όμοια υπάρχουν και οι τελεστές:

### Τελεστής

$k -= m$

$k *= m$

$k /= m$

$k \% = m$

### Σημαίνει

$k = k-m$

$k = k*m$

$k = k/m$

$k = k\%m$  (μόνο για ακεραίους)

## Προτεραιότητες

Τελεστής	Προτεραιότητα	Προσεταιριστικότητα
++ --	Υψηλή	Από δεξιά προς αριστερά
* / %		Από αριστερά προς δεξιά
+ -		Από αριστερά προς δεξιά
= += -= *= /= %=	Χαμηλή	Από δεξιά προς αριστερά

## Είσοδος - έξοδος μεταβλητών

Οι τιμές των μεταβλητών μπορούν να εμφανιστούν στην οθόνη με την εντολή `cout`. Πχ.

```
cout << x ;
```

Μηνύματα και μεταβλητές μπορεί να εμφανιστούν από την ίδια εντολή `cout`:

```
cout << "Η τιμή είναι: " << x;
```

## Είσοδος - έξοδος μεταβλητών

Εάν θέλουμε να αλλάξουμε γραμμή θα πρέπει να χρησιμοποιήσουμε το χαρακτήρα διαφυγής `\n`

```
cout << x << "\n";
```

Ο χαρακτήρας `'\n'` έχει το ειδικό όνομα `endl` που μπορούμε να χρησιμοποιήσουμε:

```
cout << x << endl;
```

Εάν θέλουμε να εμφανίσουμε δύο μεταβλητές στην ίδια γραμμή θα πρέπει να αφήσουμε κενό ανάμεσα τους:

```
cout << x << " " << y << endl;
```

## Είσοδος - έξοδος μεταβλητών

Για να εισάγουμε μεταβλητές από το πληκτρολόγιο χρησιμοποιούμε την εντολή cin:

```
cin >> x;
```

ή για περισσότερες μεταβλητές:

```
cin >> x >> y >> z;
```

Η εντολή cin χρησιμοποιείται συνήθως μαζί με μια εντολή cout:

```
cout << "Εισάγετε τις συντεταγμένες ";
cin >> x >> y >> z;
```

37

## Παράδειγμα #6

Κατασκευάστε πρόγραμμα το οποίο θα υπολογίζει τη συνολική επιφάνεια και τον όγκο κυλίνδρου όταν δίνεται η ακτίνα βάσης  $r$  και το ύψος  $h$ .

Υπενθύμιση:

Επιφάνεια:  $S = 2\pi rh + 2\pi r^2$

Όγκος:  $V = \pi r^2 h$

38

## Παράδειγμα #6

```
#include <iostream>
#include <cmath>
using namespace std;

int main ( )
{
    double r, h;
    double pi;
    double e, v;

    cout << "Εισάγετε ακτίνα και ύψος ";
    cin >> r >> h;

    pi = acos(-1.0);
    e = 2*pi*r*h + 2*pi*r*r;
    v = pi*r*r*h;

    cout << "Το εμβαδόν είναι " << e << endl;
    cout << "Ο όγκος είναι " << v << endl;
}
```

39

## Παράδειγμα #7

Κατασκευάστε πρόγραμμα το οποίο θα μετατρέπει ένα δεδομένο αριθμό δευτερολέπτων σε ώρες, λεπτά και δευτερόλεπτα.

40

## Παράδειγμα #7

```
#include <iostream>
using namespace std;
int main ( )
{
    int sec, min, hour, ypolsec, ypolmin;

    cout << "Πόσα δευτερόλεπτα ? ";
    cin >> sec;

    min = sec / 60;
    ypolsec = sec % 60;

    hour = min / 60 ;
    ypolmin = min % 60 ;

    cout << sec      << " δευτερόλεπτα είναι: " << endl;
    cout << hour     << " ώρες " << endl;
    cout << ypolmin  << " λεπτά " << endl;
    cout << ypolsec  << " δευτερόλεπτα " << endl;
}
```