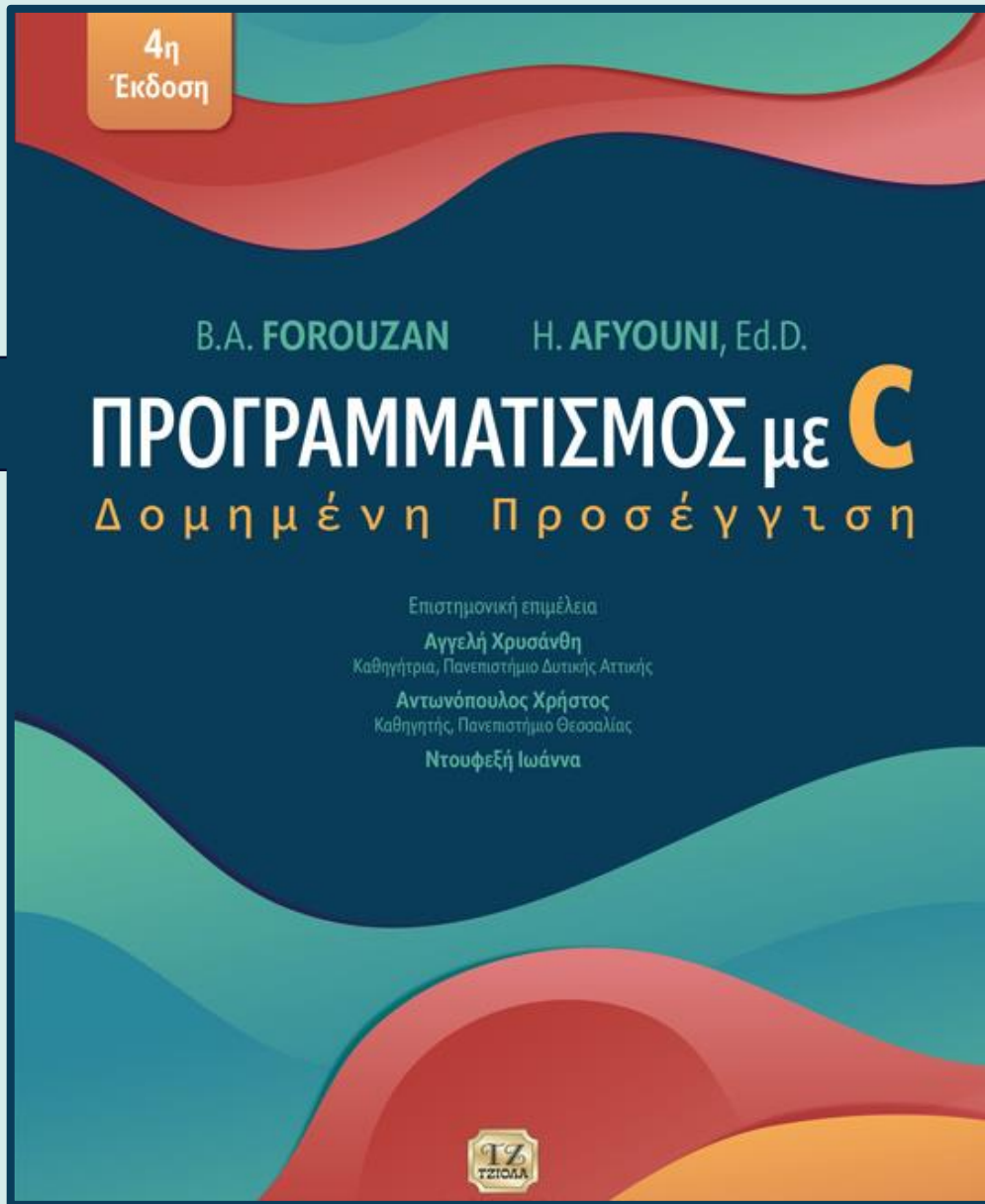




Επιστημονικές Εκδόσεις
ΤΖΙΟΛΑ

ΚΕΦΑΛΑΙΟ 2

Εισαγωγή στη Γλώσσα C

Πρωτότυπο έργο: Computer Science: A Structured Programming Approach in C, 4th ed., H. Afyouni, B.A.Forouzan. Copyright 2000, 2007, 2023 Cengage Learning, Inc.
Αποκλειστικότητα για την ελληνική γλώσσα: Εκδόσεις TZIOΛA. Copyright 2024.
Με επιφύλαξη παντός νόμιμου δικαιώματος.

ΘΕΜΑΤΑ ΚΕΦΑΛΑΙΟΥ

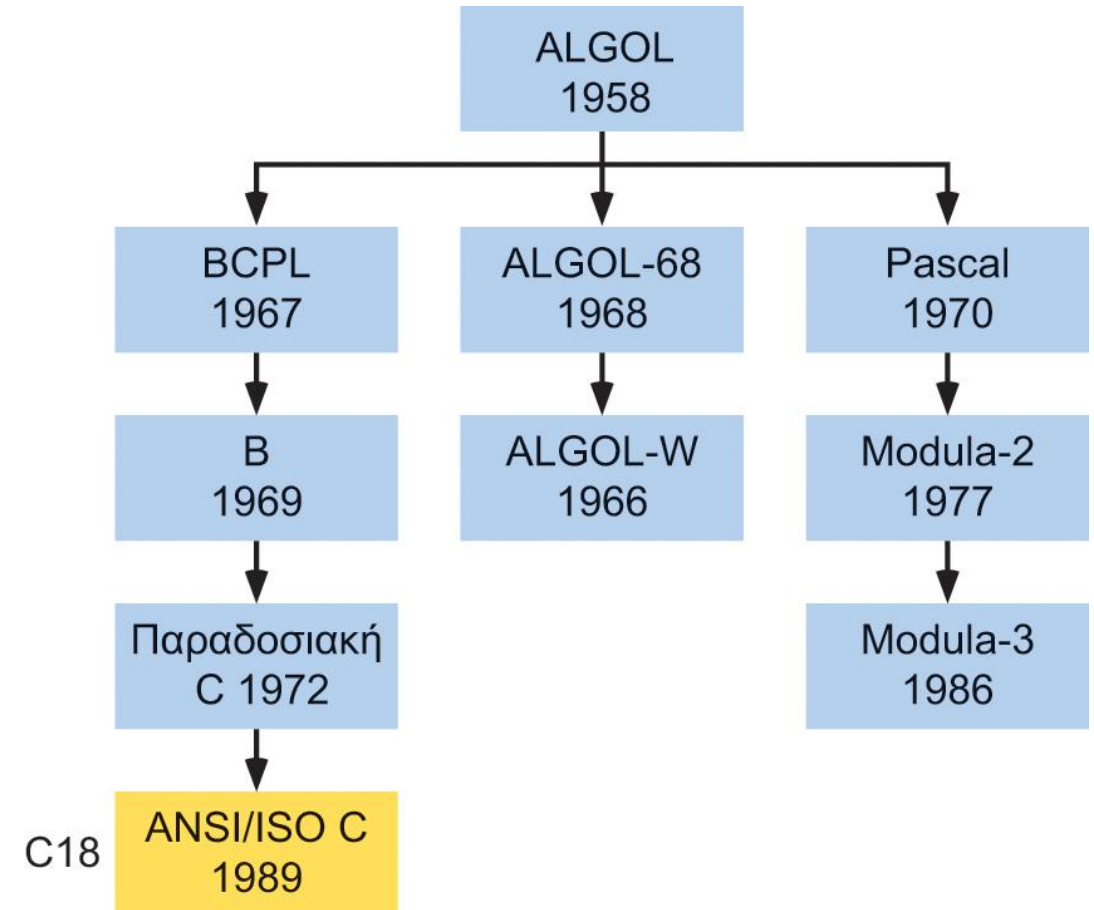
- Σύντομη αναφορά στην ανάπτυξη της γλώσσας C
- Η δομή ενός προγράμματος γραμμένου σε C
- Τα αναγνωριστικά και ο ρόλος τους σε ένα πρόγραμμα
- Οι βασικοί τύποι δεδομένων της C
- Μεταβλητές: δημιουργία και χρήση τους σε προγράμματα
- Σταθερές: δημιουργία και χρήση τους σε προγράμματα
- Απλές εντολές για είσοδο και έξοδο δεδομένων

2.1

Ιστορικό

Ιστορικό

- Στην πάροδο του χρόνου, οι γλώσσες υπολογιστών εξελίχθηκαν από γλώσσες μηχανής σε γλώσσες υψηλού επιπέδου, όπως η C
- Η C, όπως και οι περισσότερες σύγχρονες γλώσσες, προέρχεται από την ALGOL, την πρώτη γλώσσα που χρησιμοποίησε μία δομή (μπλοκ) αποτελούμενη από ακολουθία εντολών κώδικα



Ιστορικό

- Η C είναι μία γλώσσα δομημένου προγραμματισμού.
- Θεωρείται γλώσσα υψηλού επιπέδου, επειδή επιτρέπει στον προγραμματιστή να επικεντρωθεί στο πρόβλημα που αντιμετωπίζει, χωρίς να προβληματίζεται για τη «μηχανή» στην οποία θα χρησιμοποιηθεί το πρόγραμμα
 - Αν και η «ανεξαρτησία από τη μηχανή» είναι κάτι που ισχυρίζονται πολλές γλώσσες, η C φτάνει πολύ πιο κοντά από τις περισσότερες στην επίτευξη αυτού του στόχου

2.2

Προγράμματα C

Δομή ενός προγράμματος C (1/2)

- Κάθε πρόγραμμα C αποτελείται από μία ή περισσότερες εντολές προεπεξεργαστή, μία ενότητα καθολικών δηλώσεων και μία ή περισσότερες συναρτήσεις
 - Η ενότητα **καθολικών δηλώσεων** (global declaration section) βρίσκεται στην αρχή του προγράμματος
 - Η εργασία που επιτελεί ένα πρόγραμμα εκτελείται από τις **συναρτήσεις** (functions) που περιλαμβάνει: τμήματα κώδικα, τα οποία εκτελούν μία συγκεκριμένη εργασία
 - Η ενότητα **δηλώσεων** (declaration section) βρίσκεται στην αρχή της συνάρτησης
 - Η ενότητα **εντολών** (statement section) ακολουθεί την ενότητα δηλώσεων και περιέχει τις εντολές που δίνουν στον υπολογιστή οδηγίες για να κάνει κάτι.

Δομή ενός προγράμματος C (2/2)

ΣΧΗΜΑ 2-2 Η δομή
ενός προγράμματος C.

Οδηγίες προεπεξεργαστή

Δηλώσεις καθολικής εμβέλειας

```
int main (void)  
{
```

Δηλώσεις τοπικής εμβέλειας

Εντολές

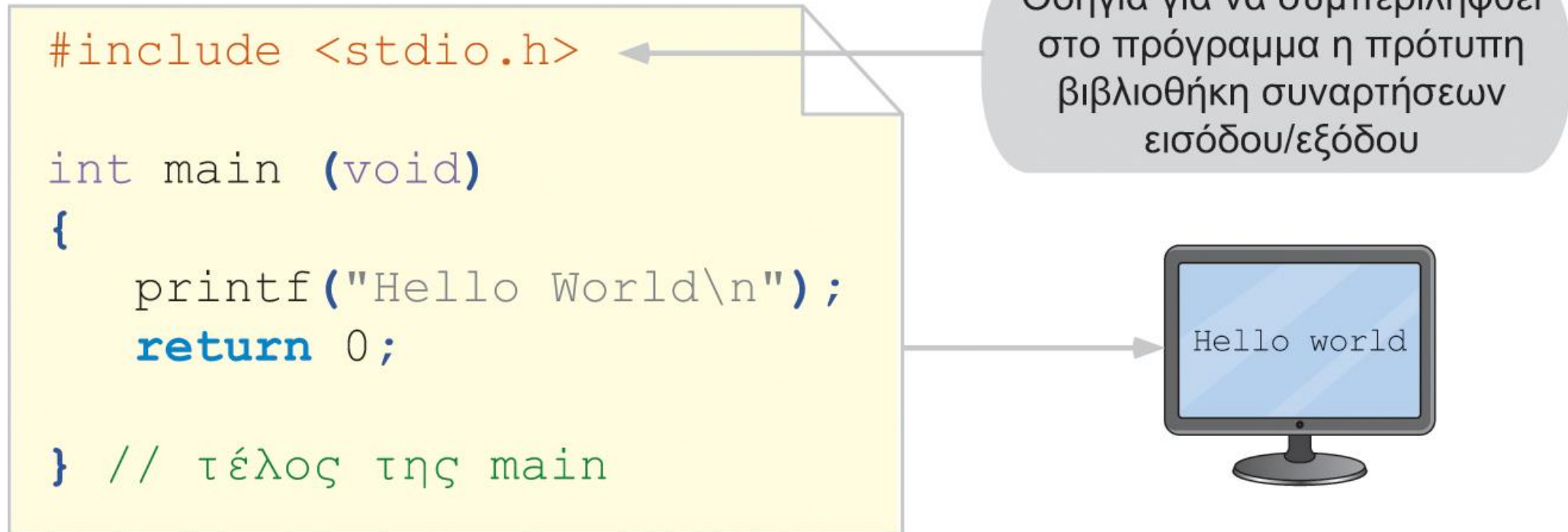
```
} // τέλος της main
```

Άλλες απαραίτητες συναρτήσεις.

Το κλασικό, πρώτο πρόγραμμα C (1/3)

- Το πρώτο πρόγραμμα που κατά παράδοση γράφουν όλοι όσοι μαθαίνουν τη C είναι πολύ απλό
 - Έχει μόνο μία οδηγία για τον προεπεξεργαστή και δεν περιλαμβάνει καθολικές δηλώσεις, ούτε τοπικούς ορισμούς
 - Ο σκοπός του είναι απλώς να «εκτυπώσει» έναν χαιρετισμό προς τον χρήστη
 - το «εκτυπώσει» σημαίνει να εμφανίσει στην οθόνη

Το κλασικό, πρώτο πρόγραμμα C (2/3)



ΣΧΗΜΑ 2-3 Το πρόγραμμα χαιρετισμού.

Το κλασικό, πρώτο πρόγραμμα C (3/3)

- Εντολές (οδηγίες) για τον προεπεξεργαστή
 - Τοποθετούνται στην αρχή του προγράμματος
 - Ξεκινούν με ένα σύμβολο # (κανόνας σύνταξης της C)
 - Π.χ., η **#include** λέει στον μεταγλωττιστή να συμπεριλάβει στο πρόγραμμα το αρχείο της πρότυπης βιβλιοθήκης συναρτήσεων εισόδου/εξόδου
- Το εκτελέσιμο τμήμα του προγράμματος ξεκινά με τη συνάρτηση **main**
 - Η **λίστα παραμέτρων** περιέχει απλώς το κείμενο του μηνύματος που θέλουμε να εμφανιστεί
 - Το κείμενο περικλείεται σε εισαγωγικά ("...") επειδή πρέπει να αντιμετωπιστεί ως συμβολοσειρά

Σχόλια (1/4)

- **Σχόλια** (comments) είναι οι επεξηγήσεις που παρεμβάλλει ο προγραμματιστής σε καίρια σημεία του κώδικα, προκειμένου να γίνει ευκολότερα κατανοητή η λογική του
 - Ένα σχόλιο πολλαπλών γραμμών (σχόλιο τύπου μπλοκ) χρησιμοποιείται όταν το κείμενό του εκτείνεται σε περισσότερες από μία γραμμές
 - Χρησιμοποιούνται ειδικά διακριτικά σύμβολα για την επισήμανση της αρχής και του τέλους ενός σχολίου πολλαπλών γραμμών (ένας ή περισσότεροι συμβολικοί χαρακτήρες που έχουν ειδική σημασία, κατανοητή από τον μεταγλωττιστή)

Σχόλια (2/4)

```
/* Αυτό είναι ένα σχόλιο σε μορφή μπλοκ πολλαπλών γραμμών
   το οποίο εκτείνεται σε δύο γραμμές. */

/*
** Μία ευρέως διαδεδομένη πρακτική είναι να τοποθετούνται τα σύμβολα
** έναρξης και τερματισμού σχολίου μόνα τους, το καθένα σε μία γραμμή.
** Το κείμενο του σχολίου γράφεται σε μία ή περισσότερες επόμενες γραμμές.
** Ορισμένοι προγραμματιστές τοποθετούν ένα ζεύγος αστερίσκων στην αρχή
** κάθε γραμμής, για να κάνουν τα σχόλια πιο ευδιάκριτα.
** */
```

ΣΧΗΜΑ 2-4 Παραδείγματα σχολίων πολλαπλών γραμμών.

Σχόλια (3/4)

- Σχόλια γραμμής (line comment): χρησιμοποιούνται για σύντομες εξηγήσεις και περιορίζονται σε μία μόνο γραμμή
 - Για την επισήμανσή τους χρησιμοποιείται ένα ζεύγος χαρακτήρων πλάγιας καθέτου (//)

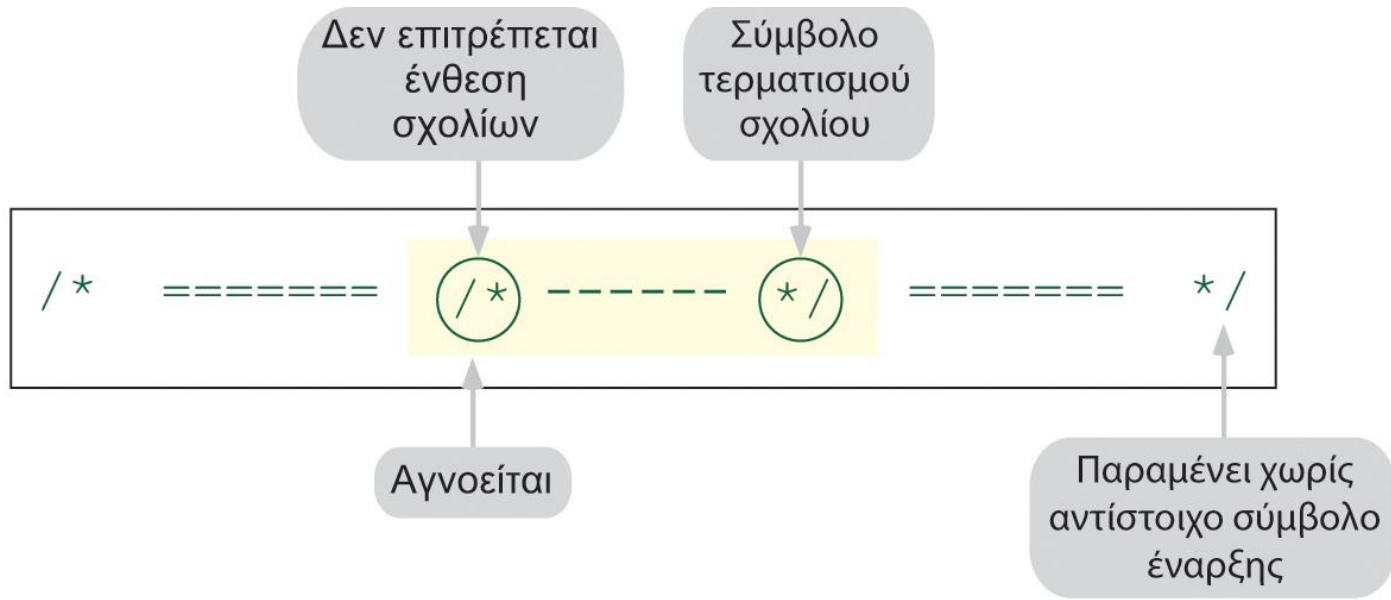
// Σχόλιο που καταλαμβάνει ολόκληρη γραμμή

a = 5; // Σχόλιο που καταλαμβάνει μέρος της γραμμής

ΣΧΗΜΑ 2-5 Παραδείγματα σχολίων γραμμής.

Σχόλια (4/4)

- Δεν επιτρέπεται η ένθεση σχολίων μέσα σε άλλα σχόλια!



Σχόλια (5/4)

- Ο μεταγλωττιστής αναφέρει σφάλμα όταν συναντά ένθετα σχόλια

```
1  /*
2  This a nested comment block which results in an error
3  /* nested comment block */
4  end of of comment block
5  */
6
7  #define <stdio,h>
8  int main()
9  {
10     printf("Helo World\n");
11     return 0;
12 } // main
```

Resources | Compile Log | Debug | Find Results | Close

Message

[Error] unknown type name 'end'

[Error] expected '=', ',', ';', 'asm' or '__attribute__' before 'of'

[Error] unknown type name 'of'

[Error] macro names must be identifiers

Το κλασικό πρόγραμμα χαιρετισμού

- Κάθε πρόγραμμα ξεκινά με ένα περιγραφικό σχόλιο τεκμηρίωσης, το οποίο εξηγεί τον σκοπό του προγράμματος
 - Αυτό αποτελεί καλή πρακτική προγραμματισμού και συνάδει με το επαγγελματικό στυλ γραφής κώδικα
 - Ανατρέξτε στο πρόγραμμα 2-1

2.3

Αναγνωριστικά

Αναγνωριστικά (1/2)

- Τα **αναγνωριστικά** επιτρέπουν στους προγραμματιστές να δίνουν ονόματα στα δεδομένα και σε άλλα αντικείμενα των προγραμμάτων τους
- Κάθε αντικείμενο το οποίο χαρακτηρίζεται με ένα αναγνωριστικό αποθηκεύεται σε μία συγκεκριμένη (μοναδική) διεύθυνση μνήμης
 - Ο κύριος σκοπός των αναγνωριστικών είναι η συμβολική αναπαράσταση των θέσεων αποθήκευσης των δεδομένων στην κύρια μνήμη του υπολογιστή
- Τα καλά αναγνωριστικά είναι περιγραφικά αλλά σύντομα και συχνά περιέχουν συντομογραφίες

Αναγνωριστικά (2/2)

ΠΙΝΑΚΑΣ 2-1 Κανόνες για την ονομασία αναγνωριστικών

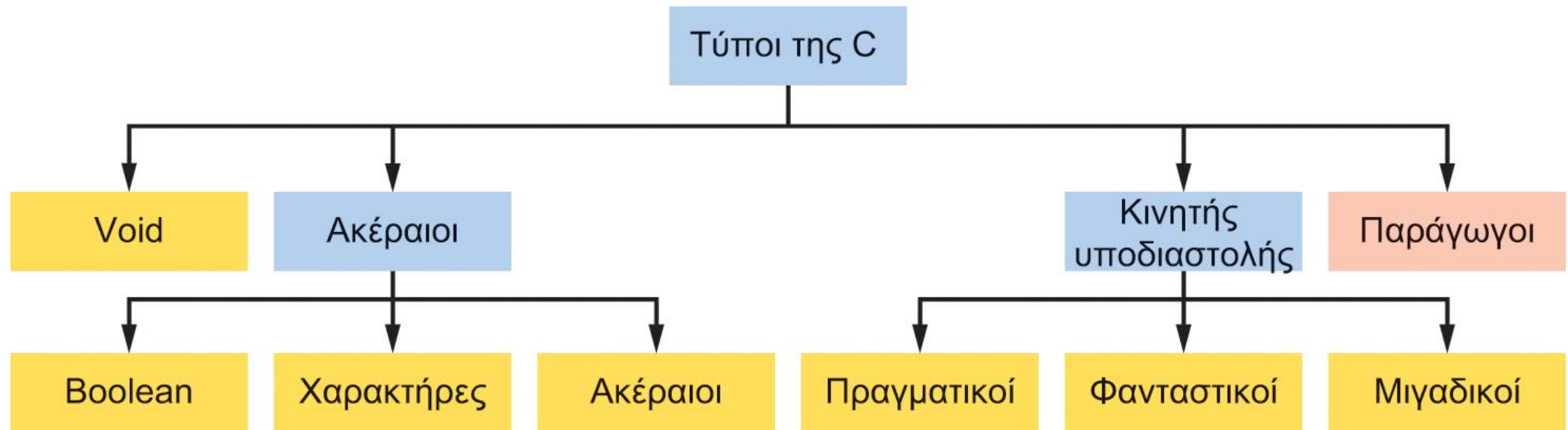
1. Ο πρώτος χαρακτήρας πρέπει να είναι αλφαβητικός ή κάτω παύλα.
 2. Μπορούν να χρησιμοποιηθούν μόνο αλφαβητικοί χαρακτήρες, ψηφία, ή κάτω παύλες.
 3. Σε ένα αναγνωριστικό θεωρούνται σημαντικοί οι πρώτοι 63 χαρακτήρες.
 4. Δεν επιτρέπονται αναγνωριστικά ίδια με δεσμευμένες λέξεις της γλώσσας.
-

2.4

Τύποι

Τύποι

- Ένας **τύπος δεδομένων** (data type) ορίζει ένα σύνολο τιμών και ένα σύνολο πράξεων που μπορούν να εφαρμοστούν σε αυτές τις τιμές



ΣΧΗΜΑ 2-7 Βασικοί και παράγωγοι τύποι δεδομένων.

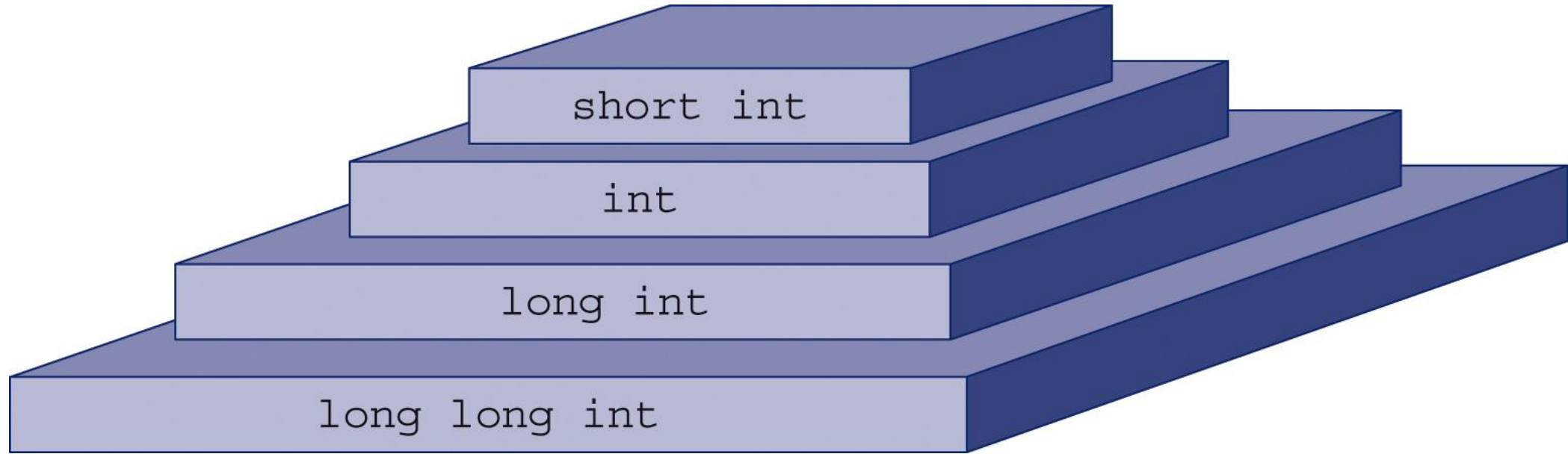
Ο τύπος void

- Ο τύπος void, που ορίζεται με τη δεσμευμένη λέξη **void**, έχει τη σημασία της «απουσίας τιμής»
- Αν και η απουσία τιμής φαίνεται ασυνήθιστη, στο πλαίσιο του προγραμματισμού ο void είναι ένας πολύ χρήσιμος τύπος δεδομένων. Για παράδειγμα, χρησιμοποιείται για να δηλώσει ότι
 - Μία συνάρτηση δεν έχει παραμέτρους
 - Μία συνάρτηση δεν έχει επιστρεφόμενη τιμή
 - Επιπλέον, μπορεί να χρησιμοποιείται για τον ορισμό «γενικευμένων» δεικτών, ώστε να δείχνουν σε δεδομένα οποιουδήποτε τύπου

Ακέραιοι τύποι

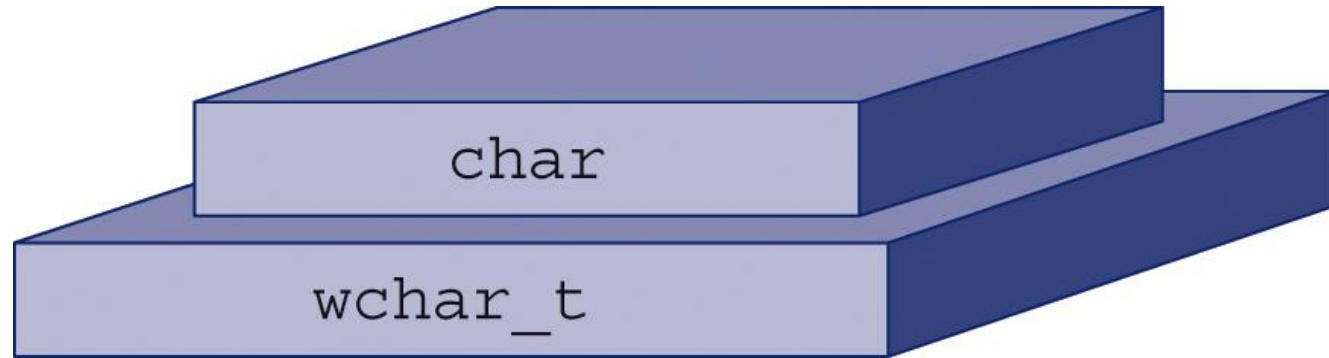
- Η γλώσσα C διαθέτει τρεις τύπους για «ακέραιο» δεδομένα
 - **bool**: μπορεί να αναπαραστήσει μόνο τις δύο λογικές τιμές, δηλαδή true (αληθές) και false (ψευδές)
 - **char** (χαρακτήρας): χαρακτήρας είναι οποιαδήποτε τιμή που μπορεί να αναπαρασταθεί στο σύνολο χαρακτήρων (στο αλφάβητο) του υπολογιστή
 - Το πρότυπο C παρέχει δύο τύπους για χαρακτήρες: char και wchar_t
 - **int** (ακέραιος): για αριθμούς χωρίς δεκαδικό μέρος
 - Η C υποστηρίζει τέσσερα διαφορετικά μεγέθη για δεδομένα ακέραιου τύπου: short int, int, long int και long long int

Ακέραιοι τύποι



ΣΧΗΜΑ 2-9 Τύποι ακέραιων αριθμών.

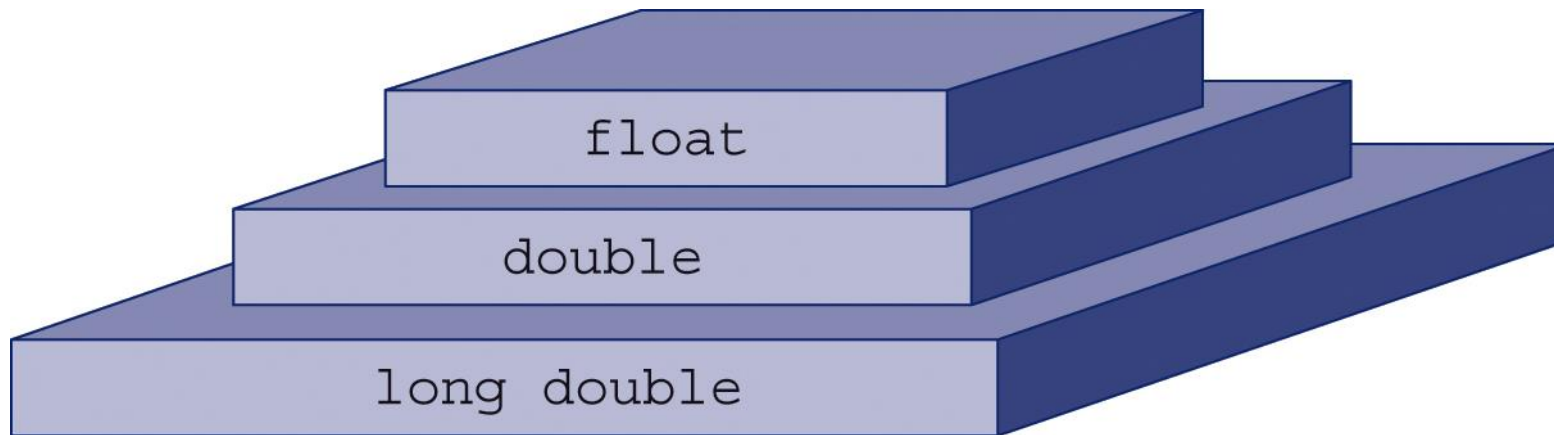
Τύποι χαρακτήρων



Τύποι πραγματικών αριθμών

- Τρεις τύποι για πραγματικούς αριθμούς (τιμές κινητής υποδιαστολής), διακρίνονται βάσει μεγέθους:

`sizeof (float) ≤ sizeof (double) ≤ sizeof (long double)`



Τύποι φανταστικών και μιγαδικών αριθμών

- Τύποι για φανταστικούς αριθμούς, διακρίνονται βάσει μεγέθους:
 - float imaginary
 - double imaginary
 - long double imaginary
- Τύποι για μιγαδικούς αριθμούς, διακρίνονται βάσει μεγέθους:
 - float complex
 - double complex
 - long long complex

Σύνοψη των τύπων δεδομένων

ΠΙΝΑΚΑΣ 2-4 Σύνοψη τύπων δεδομένων

ΚΑΤΗΓΟΡΙΑ	ΤΥΠΟΣ	ΥΛΟΠΟΙΗΣΗ ΣΤΗ C
Void	Void	void
Ακέραιοι	Λογικές τιμές	bool
	Χαρακτήρες	char, wchar_t
	Ακέραιες τιμές	short int, int, long int, long long int
Πραγματικοί αριθμοί	Τιμές κινητής υποδιαστολής, μονής και διπλής ακρίβειας	float, double, long double
	Φανταστικοί	float imaginary, double imaginary, long double imaginary
	Μιγαδικοί	float complex, double complex, long double complex

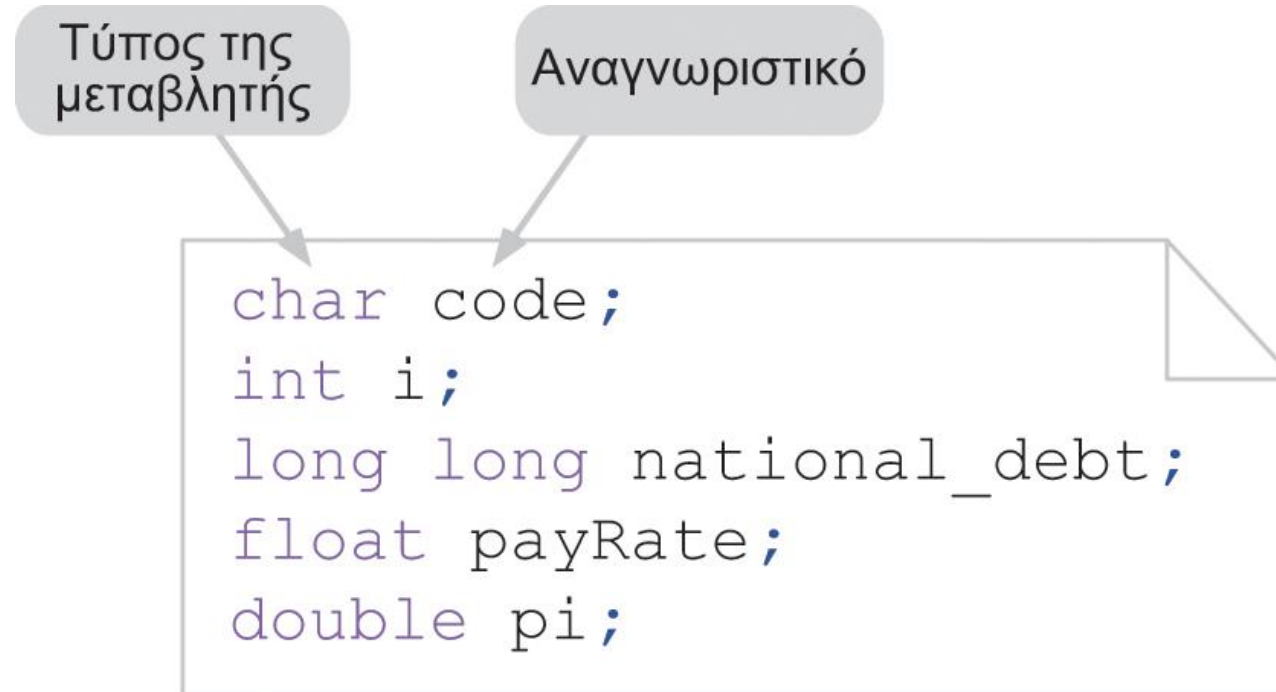
2.5

Μεταβλητές

Μεταβλητές

- Οι **μεταβλητές** (variables) είναι θέσεις μνήμης, για τις οποίες ο προγραμματιστής ορίζει ένα όνομα και έναν τύπο δεδομένων όπως, π.χ., char ή int
 - Ο **τύπος δεδομένων** καθορίζει τις τιμές που μπορεί να περιέχει μία μεταβλητή και τις λειτουργίες ή πράξεις που μπορούν να χρησιμοποιούνται για χειρισμό των τιμών της.

Μεταβλητές



Δήλωση μεταβλητών

- Κάθε μεταβλητή που χρησιμοποιείτε σε ένα πρόγραμμα πρέπει να δηλώνεται και να ορίζεται
 - Στη C, η **δήλωση** (declaration) χρησιμοποιείται για την ονομασία ενός αντικειμένου, όπως μία μεταβλητή
 - Ο **ορισμός** (definition) χρησιμοποιείται για τη δημιουργία του αντικειμένου
 - Με μία εξαίρεση, μία μεταβλητή δηλώνεται και ορίζεται ταυτόχρονα
- Ο τύπος μιας μεταβλητής μπορεί να είναι οποιοσδήποτε από τους τύπους δεδομένων της C, δηλαδή χαρακτήρας, ακέραιος ή πραγματικός
 - Μόνη εξαίρεση: μία μεταβλητή δεν μπορεί να είναι τύπου void

Δήλωση μεταβλητών

ΔΗΛΩΣΗ	ΟΡΙΣΜΟΣ	ΠΕΡΙΓΡΑΦΗ
bool	fact;	
short	maxItems;	// Χρήση κεφαλαίου χαρακτήρα για διαχωρισμό λέξεων
long	Long_national_debt;	// Χρήση κάτω παύλας για διαχωρισμό λέξεων
float	payRate;	// Χρήση κεφαλαίου χαρακτήρα για διαχωρισμό λέξεων
double	tax;	
float	complex voltage;	
char	code, kind;	// Προβληματικό στυλ γραφής κώδικα // Προτιμάται ο ορισμός κάθε μιας σε ξεχωριστή εντολή.
int	a, b;	// Προβληματικό στυλ γραφής κώδικα. Δείτε κείμενο. // Προτιμάται ο ορισμός κάθε μιας σε ξεχωριστή εντολή.

Αρχικοποίηση μεταβλητών

- Μπορούμε να αρχικοποιήσουμε μία μεταβλητή τη στιγμή που τη δηλώνουμε, συμπεριλαμβάνοντας μία τιμή αρχικοποίησης
 - Όταν υπάρχει, η **τιμή αρχικοποίησης** (initializer) καθορίζει την αρχική τιμή που θα περιέχει η μεταβλητή
- Για την αρχικοποίηση μιας μεταβλητής όταν ορίζεται, το αναγνωριστικό ακολουθείται από ένα σύμβολο ίσον (=), το οποίο είναι ο τελεστής ανάθεσης τιμής, και στη συνέχεια την τιμή αρχικοποίησης, η οποία είναι η αρχική τιμή της μεταβλητής κατά την έναρξη της συνάρτησης ή του προγράμματος

Αρχικοποίηση μεταβλητών

```
char    code    = 'B';  
int     i       = 14;  
long long nat1_debt = 100000000000000;  
float   payRate  = 14.25;  
double  pi       = 3.1415926536;
```

Πρόγραμμα

```
B code  
14 i  
100000000000000 nat1_debt  
14.25 payRate  
3.1415926536 pi
```

Μνήμη

Όταν ορίζεται μία μεταβλητή, δεν αρχικοποιείται.
Η καλύτερη πρακτική είναι να αρχικοποιείτε κάθε μεταβλητή που χρειάζεται προκαθορισμένα δεδομένα κατά την έναρξη της συνάρτησης ή του προγράμματος.

2.6

Σταθερές

Σταθερές

- Μία **σταθερά** (constant) είναι μία τιμή δεδομένων που δεν μπορεί να αλλάξει κατά τη διάρκεια της εκτέλεσης ενός προγράμματος
- Όπως και οι μεταβλητές, οι σταθερές συσχετίζονται με έναν τύπο δεδομένων
 - Σταθερές λογικών (boolean) τιμών, χαρακτήρων, ακεραίων, πραγματικών, μιγαδικών και συμβολοσειρών

Αναπαράσταση σταθερών

- Σταθερές λογικών τιμών
 - Ο τύπος δεδομένων Boolean μπορεί να λάβει μόνο δύο τιμές: true (αληθές) και false (ψευδές). Επομένως, χρησιμοποιούνται μόνο δύο σύμβολα για την αναπαράσταση μιας σταθεράς τύπου Boolean: 0 (false) και 1 (true)
- Σταθερές χαρακτήρων
 - Μία σταθερά χαρακτήρα περικλείεται μεταξύ δύο αποστρόφων (' ')
 - Οι μη εκτυπώσιμοι χαρακτήρες φέρουν ως πρόθεμα μία πλάγια κάθετο, \ (backslash, χαρακτήρας διαφυγής)

Αναπαράσταση σταθερών

- Σταθερές ακέραιων τιμών
 - Παρόλο που οι ακέραιοι αποθηκεύονται πάντα στη δυαδική τους μορφή, γράφονται όπως χρησιμοποιούνται στην καθημερινή ζωή
 - Δηλαδή, η τιμή 15 γράφεται απλά ως 15

ΑΝΑΠΑΡΑΣΤΑΣΗ	ΤΙΜΗ	ΠΡΟΕΠΙΛΕΓΜΕΝΟΣ ΤΥΠΟΣ
+ 123	123	int
-378	-378	int
-32271L	-32,271	long int
76542LU	76,542	unsigned long int
12789845LL	12,789,845	long long int

Αναπαράσταση σταθερών

- Σταθερές πραγματικών τιμών
 - Χρησιμοποιείται ως προεπιλεγμένος ο τύπος double
 - Εάν θέλετε ο τύπος δεδομένων μιας σταθεράς να είναι float ή long double, πρέπει να το προσδιορίσετε ρητά στον κώδικα

ΑΝΑΠΑΡΑΣΤΑΣΗ	ΤΙΜΗ	ΤΥΠΟΣ
0.	0.0	double
.0	0.0	double
2.0	2.0	double
3.1416	3.1416	double
-2.0f	-2.0	float
3.1415926536L	3.1415926536	long double

Αναπαράσταση σταθερών

- Σταθερές μιγαδικών τιμών

- Οι μιγαδικοί αριθμοί περιλαμβάνουν πραγματικό και φανταστικό μέρος, τα οποία διαχωρίζονται με ένα σύμβολο συν (+)
- Τα δύο μέρη μιας μιγαδικής σταθεράς πρέπει να έχουν την ίδια ακρίβεια

ΑΝΑΠΑΡΑΣΤΑΣΗ	ΤΙΜΗ	ΤΥΠΟΣ
12.3 + 14.4 * I	$12.3 + 14.4 * (-1)^{1/2}$	double complex
14F + 16F * I	$14 + 16 * (-1)^{1/2}$	float complex
1.4736L + 4.567561 * I	$1.4736 + 4.56756 * (-1)^{1/2}$	long double complex

Αναπαράσταση σταθερών

- Σταθερές συμβολοσειράς

- Μία σταθερά συμβολοσειράς είναι μία ακολουθία μηδέν ή περισσότερων χαρακτήρων που περικλείονται σε εισαγωγικά (" ")

`" "`

`// Κενή συμβολοσειρά`

`"h"`

`// Συμβολοσειρά ενός χαρακτήρα`

`"Καλημέρα!"`

`"Hello World\n"`

`// Συμβολοσειρά με χαρακτήρα νέας γραμμής`

`L"Συμβολοσειρά με χαρακτήρες πλήρους πλάτους."`

Κενός χαρακτήρας και κενή συμβολοσειρά

' \ 0 '



Κενός (null) χαρακτήρας: ένα byte του οποίου τα 8 bits έχουν τιμή 0.

" "



Κενή συμβολοσειρά: δεν περιέχει τίποτα

Χρήση σταθερών σε ένα πρόγραμμα (1/2)

- Κυριολεκτικές σταθερές
 - Ένα κυριολεκτικό ή παραστατική σταθερά είναι μία ανώνυμη σταθερά που χρησιμοποιείται για τον προσδιορισμό συγκεκριμένων δεδομένων
 - Γράφονται ως μέρος μιας εντολής χρησιμοποιώντας τις μορφές για τους διάφορους τύπους σταθερών
- Οριζόμενες σταθερές
 - Μπορούμε να ορίσουμε μία σταθερά χρησιμοποιώντας την εντολή `#define` του προεπεξεργαστή

Χρήση σταθερών σε ένα πρόγραμμα (2/2)

- Σταθερές μνήμης (memory constants)
- Μία σταθερά μνήμης χρησιμοποιεί το προσδιοριστικό τύπου **const** της C, για να υποδείξει ότι τα δεδομένα δεν μπορούν να αλλάξουν

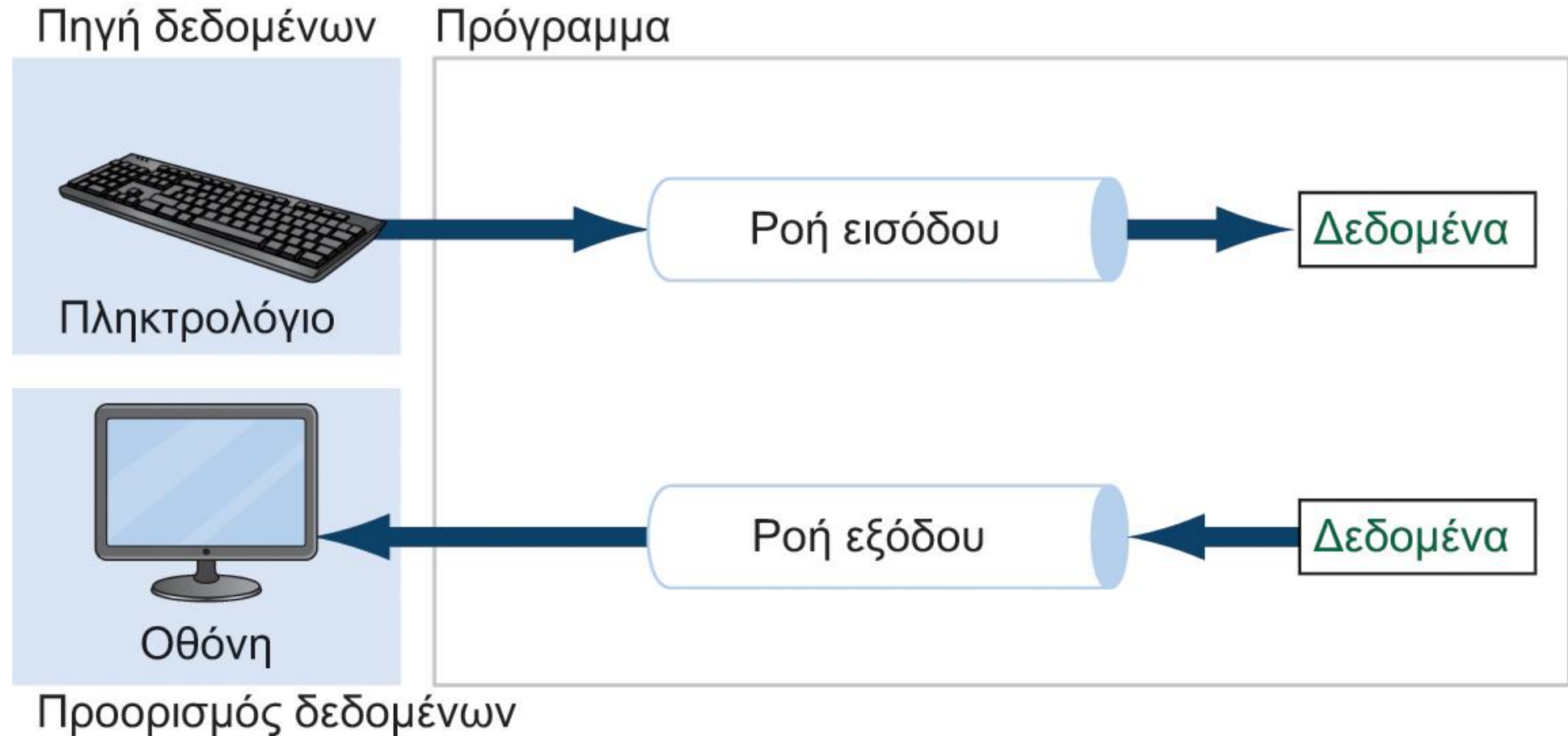
2.7

Είσοδος/Εξοδος

Ροές δεδομένων (1/2)

- Στη C, η είσοδος και η έξοδος δεδομένων πραγματοποιείται μέσω μιας **ροής** (stream)
 - Μία ροή λειτουργεί ως πηγή ή προορισμός δεδομένων
 - Μπορεί να συσχετίζεται είτε με μία φυσική συσκευή (π.χ., πληκτρολόγιο), είτε με ένα αρχείο σε κάποιο μέσο αποθήκευσης (π.χ., σκληρός δίσκος)
 - **Ροή κειμένου** (text stream): μία ακολουθία χαρακτήρων κειμένου που χωρίζονται σε γραμμές και κάθε γραμμή τερματίζεται με τον χαρακτήρα νέας γραμμής (\n).
 - **Δυαδική ροή** (binary stream): μία ακολουθία τιμών δεδομένων, όπως ακέραιοι, πραγματικοί ή μιγαδικοί αριθμοί, στη μορφή με την οποία αναπαρίστανται στη μνήμη του υπολογιστή

Ροές δεδομένων (2/2)

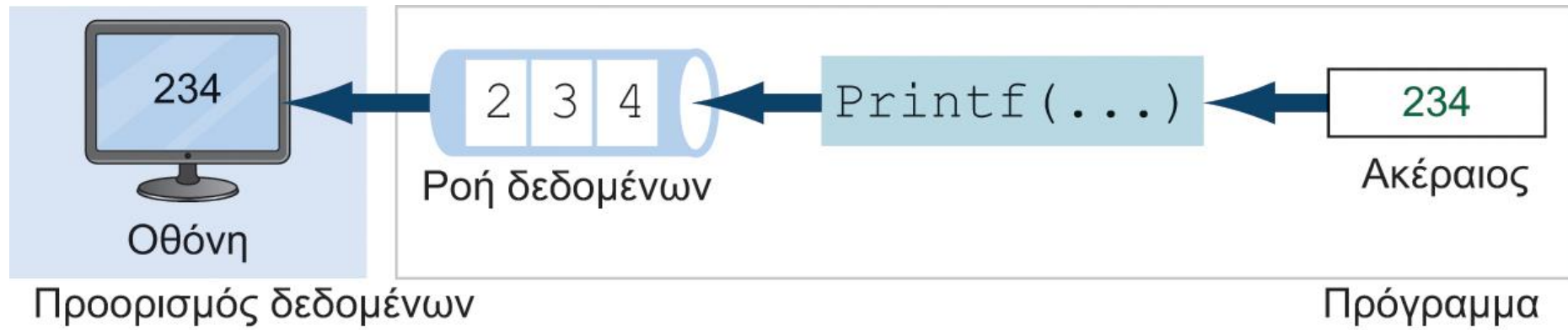


ΣΧΗΜΑ 2-15 Ροές και φυσικές συσκευές στις οποίες αντιστοιχούν.

Μορφοποίηση εισόδου/εξόδου

- Μορφοποίηση της εξόδου με τη συνάρτηση **printf**
 - Λαμβάνει ένα σύνολο τιμών δεδομένων, τις μετατρέπει σε ροή κειμένου βάσει των προδιαγραφών μορφοποίησης που περιέχονται σε μία συμβολοσειρά ελέγχου μορφοποίησης και στέλνει την παραγόμενη ροή κειμένου στην τυπική έξοδο (οθόνη)

Μορφοποίηση εισόδου/εξόδου



Μορφοποίηση εισόδου/εξόδου

- Για να καθορίσει πώς πρέπει να μορφοποιηθεί η ροή κειμένου, χρησιμοποιεί μία συμβολοσειρά ελέγχου μορφοποίησης η οποία μπορεί να περιέχει μηδέν ή περισσότερες προδιαγραφές μετατροπής

%	ΣΗΜ	Ελάχιστο εύρος	Ακρίβεια	Μέγεθος	ΚΩΔ
---	-----	-------------------	----------	---------	-----

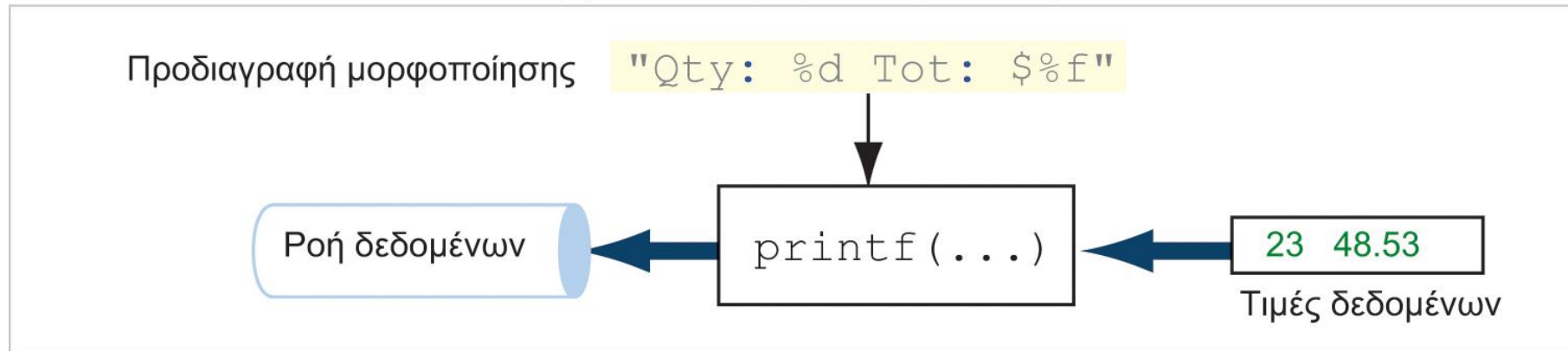
ΣΗΜ: Σήμανση, ΚΩΔ: Κωδικός

Μορφοποίηση εισόδου/εξόδου

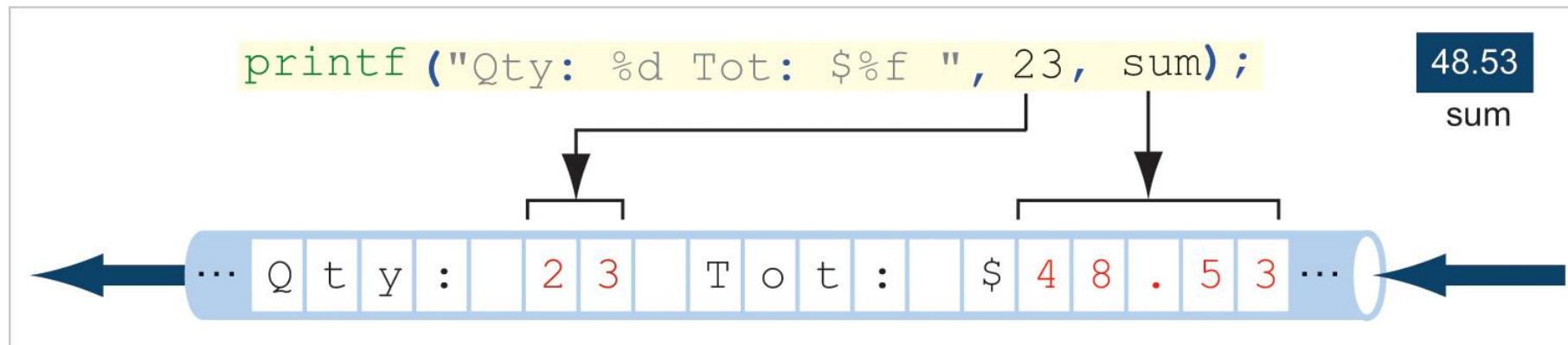
- Μορφοποίηση της εισόδου με τη συνάρτηση **scanf**
 - λαμβάνει μία ροή κειμένου από το πληκτρολόγιο, εξάγει τα δεδομένα από τη ροή, τα μορφοποιεί με βάση τις προδιαγραφές της συμβολοσειράς ελέγχου μορφοποίησης και στη συνέχεια αποθηκεύει τα δεδομένα σε προσδιοριζόμενες μεταβλητές του προγράμματος
 - είναι το αντίστροφο της συνάρτησης printf, υπό την έννοια ότι εκτελεί την αντίστροφη λειτουργία: είσοδο αντί για έξοδο

Μορφοποίηση εισόδου/εξόδου

(α) Εννοιολογική αναπαράσταση

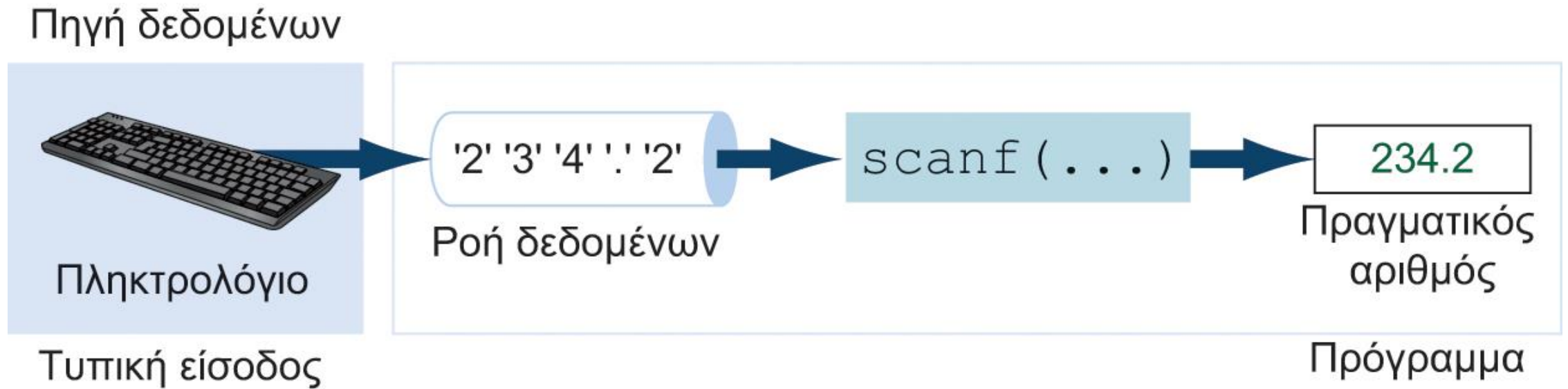


(β) Υλοποίηση



ΣΧΗΜΑ 2-17 Παράδειγμα μορφοποίησης της ροής εξόδου.

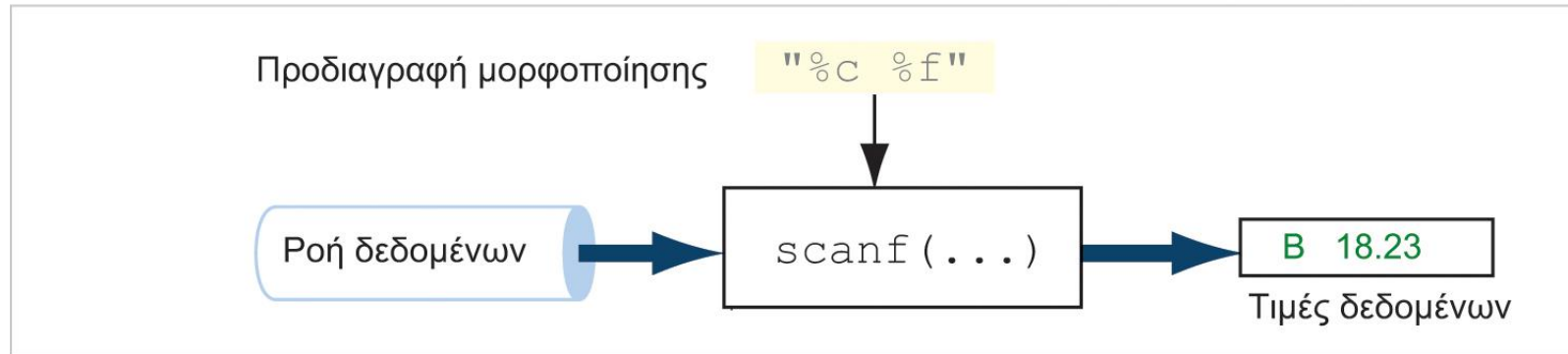
Μορφοποίηση εισόδου/εξόδου



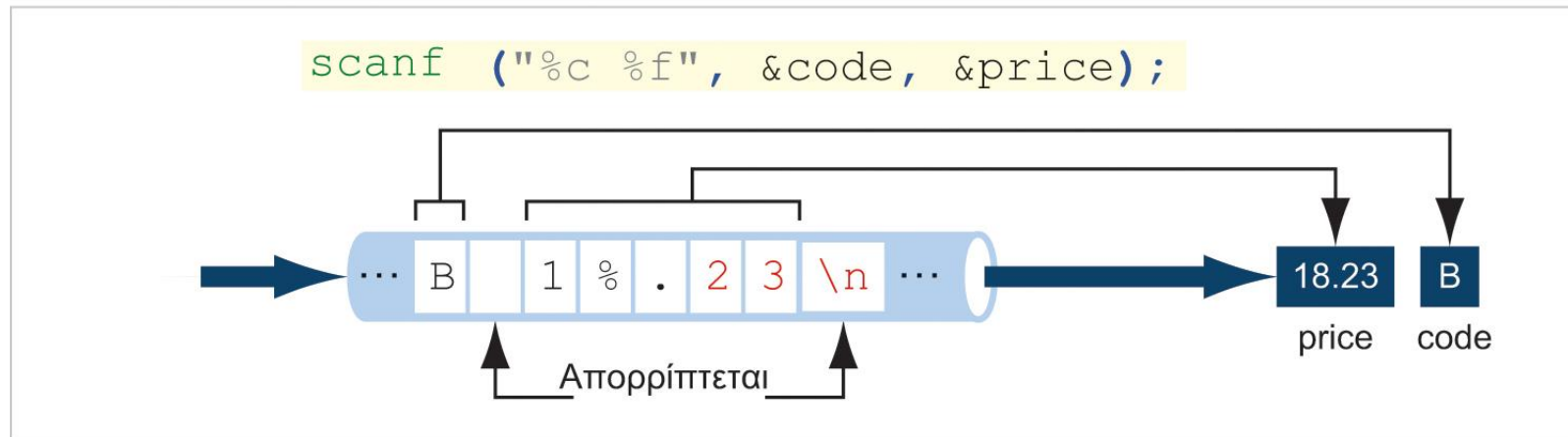
ΣΧΗΜΑ 2-19 Μορφοποίηση κειμένου από τη ροή εισόδου.

Μορφοποίηση εισόδου/εξόδου

(α) Εννοιολογική αναπαράσταση



(β) Υλοποίηση



ΣΧΗΜΑ 2-20 Μορφοποίηση δεδομένων που διαβάζονται από τη ροή εισόδου.

Σύνοψη

- Το 1972, ο Dennis Ritchie σχεδίασε τη γλώσσα C στα Bell Laboratories
- Το 1989, το Αμερικανικό Ινστιτούτο Εθνικών Προτύπων (ANSI) ενέκρινε το πρότυπο ANSI C, ενώ το 1990 εγκρίθηκε το πρότυπο ISO
- Το βασικό συστατικό ενός προγράμματος C είναι η συνάρτηση
- Κάθε συνάρτηση της C αποτελείται από δηλώσεις, ορισμούς και μία ή περισσότερες εντολές
- Μία και μόνο μία από τις συναρτήσεις ενός προγράμματος C πρέπει να ονομάζεται main
- Τα σχόλια να κάνουν πιο ευκολονόητο ένα πρόγραμμα. Ένα σχόλιο είναι μία ακολουθία χαρακτήρων που αγνοούνται από τον μεταγλωττιστή. Η C χρησιμοποιεί δύο τύπους σχολίων: πολλαπλών γραμμών και μιας γραμμής
- Τα αναγνωριστικά χρησιμοποιούνται σε μία γλώσσα για την ονομασία αντικειμένων

Σύνοψη

- Στους βασικούς τύπους της C περιλαμβάνονται ο void, οι τύποι για ακέραιους και πραγματικούς (κινητής-υποδιαστολής) αριθμούς και οι παραγόμενοι τύποι
- Ο τύπος void χρησιμοποιείται όταν η C πρέπει να ορίσει την απουσία δεδομένων
- Οι «ακέραιοι» τύποι στη C διακρίνονται περαιτέρω σε τύπους λογικών (boolean) τιμών, τύπους χαρακτήρων και τύπους ακέραιων αριθμών
- Για τον τύπο λογικών (boolean) τιμών προβλέπονται μόνο δύο τιμές: true και false. Ορίζεται με τη δεσμευμένη λέξη bool
- Ο τύπος δεδομένων χαρακτήρων χρησιμοποιεί τιμές από το τυπικό σύνολο χαρακτήρων του συστήματος, όπως ASCII ή Unicode. Προβλέπονται δύο τύποι, ένας για συμβατικούς χαρακτήρες (char) και ένας για χαρακτήρες πλήρους εύρους (w_char)

Σύνοψη

- Η C παρέχει τέσσερα διαφορετικά μεγέθη για τους ακέραιους: short int, int, long int και long long int
- Για τους πραγματικούς αριθμούς, η C παρέχει τρεις τύπους που υποστηρίζουν διαφορετικά μεγέθη: float, double και long double
- Για τους φανταστικούς αριθμούς, η C παρέχει τρεις τύπους που υποστηρίζουν διαφορετικά μεγέθη: float imaginary, double imaginary και long double imaginary
- Για τους μιγαδικούς αριθμούς, η C παρέχει τρεις τύπους που υποστηρίζουν διαφορετικά μεγέθη: float complex, double complex και long double complex
- Μία σταθερά είναι δεδομένα των οποίων η τιμή δεν μπορεί να αλλάξει
- Οι σταθερές μπορούν να γραφούν με τρεις διαφορετικούς τρόπους: ως κυριολεκτικά, ως οριζόμενες από τον χρήστη με την εντολή #define και ως σταθερές μνήμης

Σύνοψη

- Οι μεταβλητές είναι «επώνυμες» περιοχές της μνήμης που χρησιμοποιούνται για την αποθήκευση δεδομένων
- Στη C, οι μεταβλητές πρέπει να δηλωθούν και να οριστούν πριν χρησιμοποιηθούν
- Για την είσοδο δεδομένων μέσω του πληκτρολογίου και την έξοδο δεδομένων στην οθόνη, χρησιμοποιούνται οι συναρτήσεις εισόδου/εξόδου που περιλαμβάνονται στην πρότυπη βιβλιοθήκη της C
- Τα προγράμματα θα πρέπει να περιέχουν σχόλια προκειμένου να παρέχουν στον αναγνώστη πληροφορίες τεκμηρίωσης για τον κώδικα



Το παρόν συνοδευτικό έργο **παρέχεται** αποκλειστικά και μόνο στους διδάσκοντες που έχουν επιλέξει το αντίστοιχο βιβλίο μέσω του συστήματος του **Ευδόξου** ως διδακτικό σύγγραμμα για τη διδασκαλία των μαθημάτων τους και την αξιολόγηση της εκμάθησης των φοιτητών και για όσο χρονικό διάστημα διατηρείται η επιλογή του συγκεκριμένου συγγράμματος. Η **διάδοση, δημοσίευση, αναπαραγωγή ή πώληση** οπουδήποτε μέρους αυτού του έργου με οποιοδήποτε μέσο καταστρέφει την ακεραιότητα του έργου **και για σκοπούς διαφορετικούς από αυτούς για τους οποίους έχει χορηγηθεί η σχετική άδεια δεν επιτρέπεται**. Το περιεχόμενο του έργου **δεν θα πρέπει να διατίθεται** στους φοιτητές παρά μόνο όταν αυτό αναφέρεται ρητά ότι μπορεί να γίνει. Η **χρήση** του παρόντος έργου γίνεται αποκλειστικά από τον διδάσκοντα στα πλαίσια των εκπαιδευτικών καθηκόντων του και με τη χρήση των μέσων που αυτός διαχειρίζεται και έχει στη διάθεσή του από τις **επίσημες υπηρεσίες του εκπαιδευτικού ιδρύματος** όπου ανήκει, διασφαλίζοντας τη μη περαιτέρω διάδοση, δημοσίευση και αναπαραγωγή του έργου προς τρίτους. Ο διδάσκων δύναται να **προσαρμόζει** μέρος του υλικού των διαφανειών σύμφωνα με τις διδακτικές του ανάγκες, αναφερόμενος όμως πάντοτε στην αρχική πηγή αναφοράς. Το παρόν έργο προστατεύεται από την ελληνική και ευρωπαϊκή νομοθεσία περί πνευματικής ιδιοκτησίας καθώς και από τη νομοθεσία του κράτους όπου ανήκει η ξενόγλωσση πρωτότυπη έκδοση του έργου.