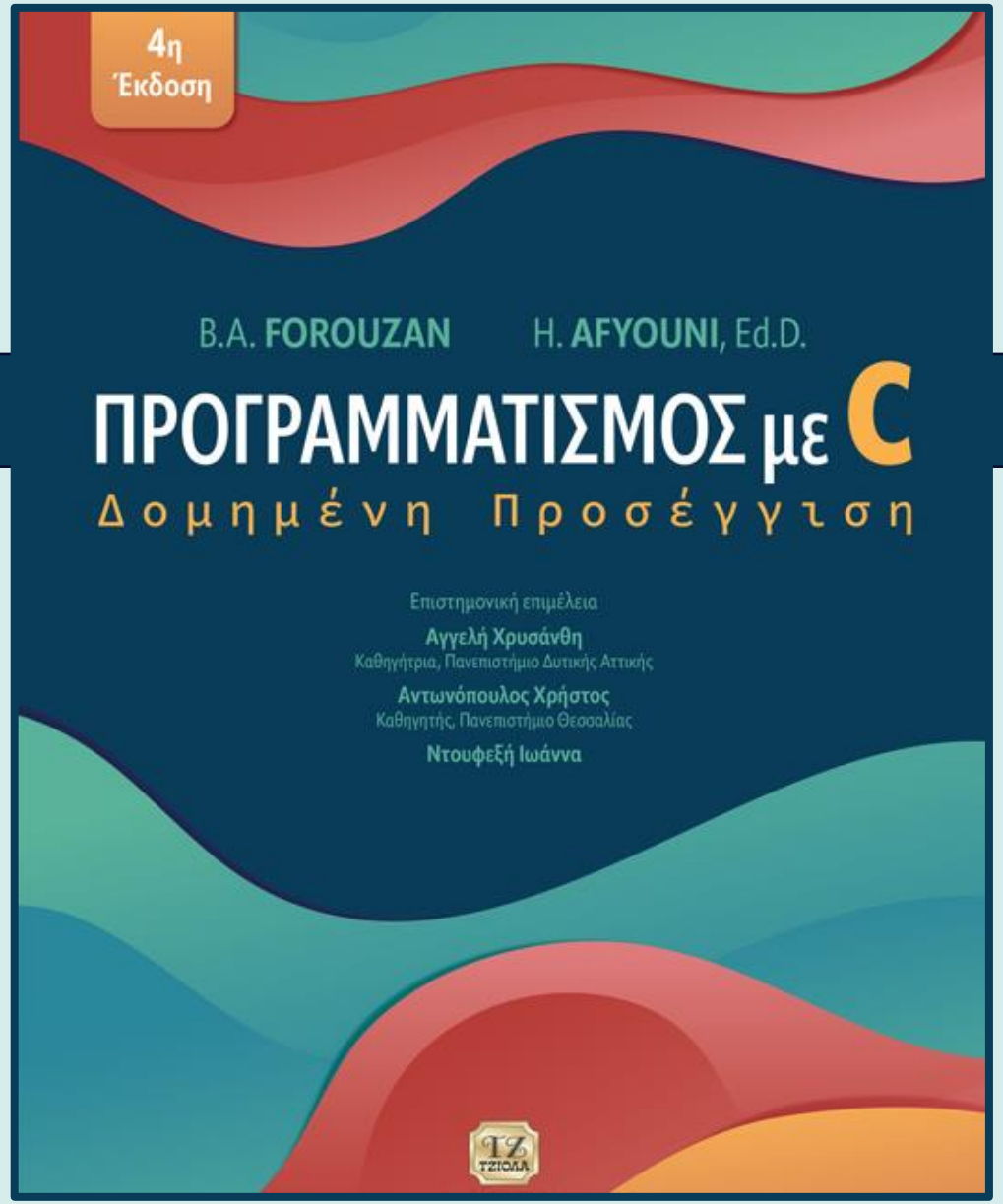


Επιστημονικές Εκδόσεις
ΤΖΙΟΛΑ

ΚΕΦΑΛΑΙΟ 4

Συναρτήσεις

ΘΕΜΑΤΑ ΚΕΦΑΛΑΙΟΥ

Χρήση διαγραμμάτων δομής για τη σχεδίαση προγραμμάτων με πολλαπλές συναρτήσεις

Σχεδίαση και υλοποίηση προγραμμάτων με περισσότερες από μία συναρτήσεις

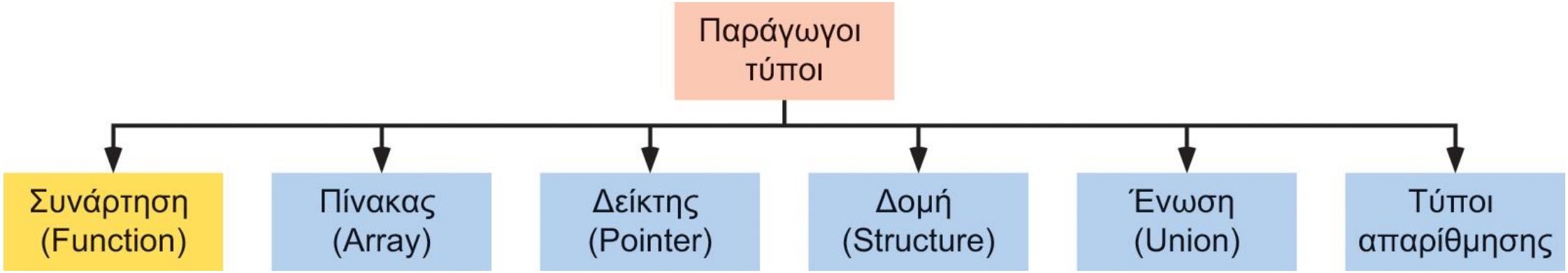
Ο ρόλος της δήλωσης, του ορισμού και της κλήσησμιας συνάρτησης

Επικοινωνία μεταξύ δύο συναρτήσεων, μέσω παραμέτρων

Οι τέσσερις βασικοί τύποι συναρτήσεων

Ο ρόλος της εμβέλειας και η η διάκριση μεταξύ καθολικής και τοπικής εμβέλειας

Σχεδίαση δομημένων προγραμμάτων



ΣΧΗΜΑ 4-1 Παράγωγοι τύποι της C.

4.1
Σχεδίαση δομημένων προγραμμάτων

Σχεδίαση δομημένων προγραμμάτων

- Δεν είναι δυνατόν να κατανοήσετε όλες τις πτυχές ενός μεγάλου προγράμματος χωρίς να τις αναγάγετε, με κάποιον τρόπο, σε απλούστερα μέρη
  - Η ανάλυση ενός πολύπλοκου προβλήματος σε μικρότερα, απλούστερα μέρη, είναι κοινή πρακτική
  - Κάθε τέτοιο μέρος προγράμματος αποκαλείται λειτουργική μονάδα, ή απλώς μονάδα (module)
  - Η διαδικασία διαχωρισμού, ή αποσύνθεσης ενός προβλήματος σε διαχειρίσιμα μέρη, αποκαλείται σχεδίαση από πάνω προς τα κάτω (top-down design)

Σχεδίαση δομημένων προγραμμάτων

- Ο τύπος συνάρτησης προκύπτει από τον τύπο της τιμής που επιστρέφει η συνάρτηση
  - Ο τύπος επιστρεφόμενης τιμής, ή τύπος επιστροφής (return type) της συνάρτησης, μπορεί να είναι οποιοσδήποτε τύπος δεδομένων, με εξαίρεση τον τύπο πίνακα

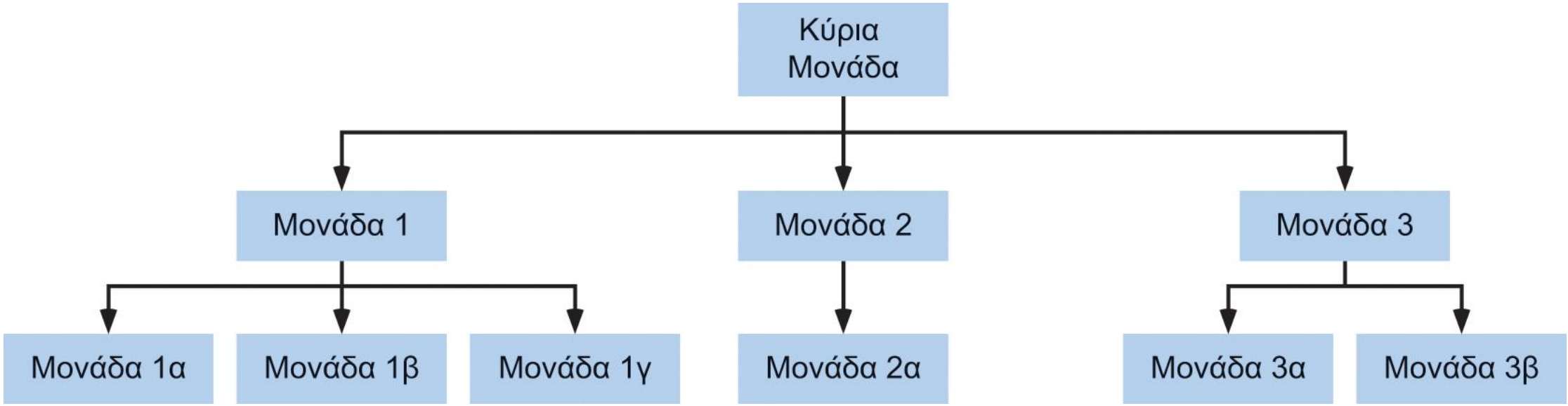
Σχεδίαση δομημένων προγραμμάτων

- Η διαδικασία διαχωρισμού-ανάλυσης αναφέρεται ως ανάλυση παραγόντων (factoring) συνεχίζεται μέχρις ότου να φτάσουμε σε υπομονάδες αποτελούμενες μόνο από στοιχειώδεις διαδικασίες που είναι άμεσα κατανοητές και δεν μπορούν να διαχωριστούν περαιτέρω
  - Η κύρια μονάδα αναφέρεται ως καλούσα μονάδα επειδή έχει (και καλεί) υπομονάδες
  - Κάθε μία από τις υπομονάδες αναφέρεται ως καλούμενη μονάδα

Σχεδίαση δομημένων προγραμμάτων

- Το πέρασμα δεδομένων σε μία συνάρτηση γίνεται χρησιμοποιώντας μία μέθοδο γνωστή ως πέρασμα παραμέτρων (parameter passing)
  - Οι παράμετροι μιας συνάρτησης περιλαμβάνονται σε μία λίστα, η οποία αποτελεί τον ορισμό των δεδομένων που περνάει στη συνάρτηση ο κώδικας που την καλεί

Σχεδίαση δομημένων προγραμμάτων



ΣΧΗΜΑ 4-2 Παράδειγμα διαγράμματος δομής.

4.2 Συναρτήσεις στη C

Συναρτήσεις στη C

- Στη γλώσσα C, η σχεδίαση από πάνω προς τα κάτω υλοποιείται χρησιμοποιώντας μία ή περισσότερες μονάδες κώδικα
  - Ένα πρόγραμμα της C αποτελείται από μία ή περισσότερες συναρτήσεις, μία (και μόνο) εκ των οποίων πρέπει να ονομάζεται **`main`**
  - Η εκτέλεση του προγράμματος ξεκινά και τελειώνει πάντοτε με τη συνάρτηση `main`, αλλά αυτή μπορεί να καλεί άλλες συναρτήσεις για την εκτέλεση ειδικών εργασιών
    - Μία καλούμενη συνάρτηση λαμβάνει τον έλεγχο από μία καλούσα συνάρτηση
    - Όταν η καλούμενη συνάρτηση ολοκληρώσει την εργασία της, επιστρέφει τον έλεγχο στην καλούσα συνάρτηση

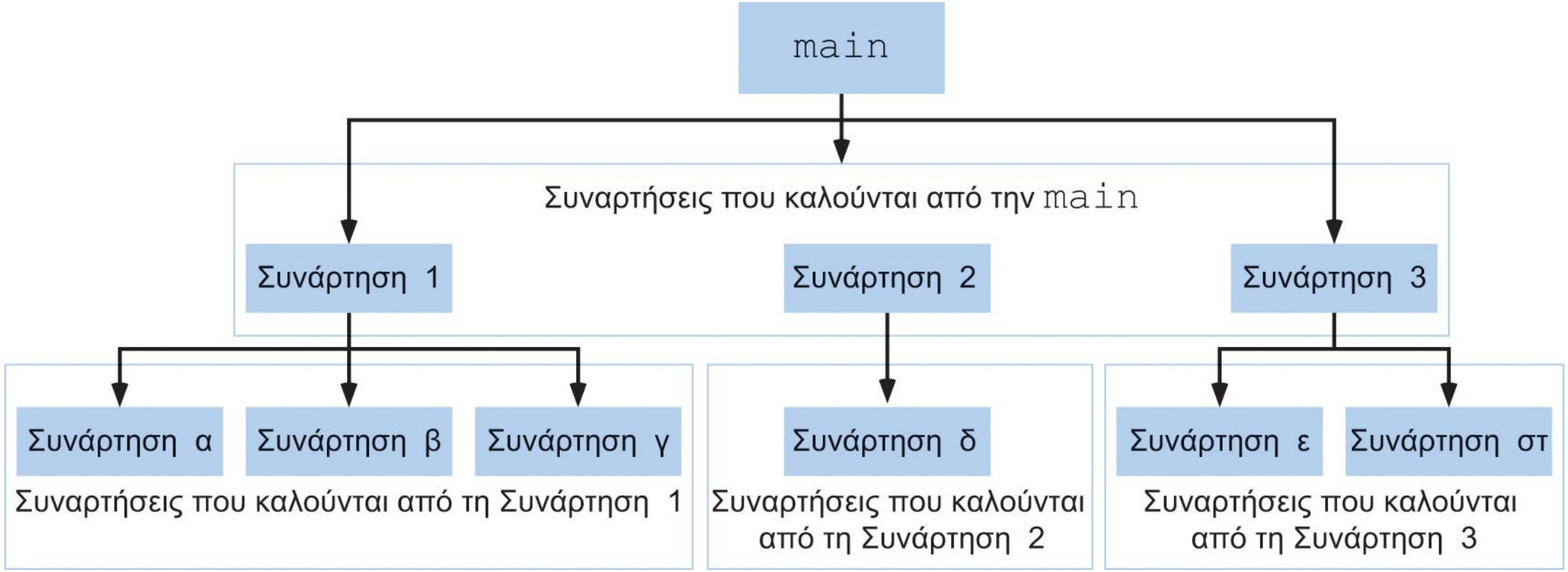
Συναρτήσεις στη C

- Η γενική μορφή ενός ορισμού συνάρτησης είναι:

```
return_type function_name (parameter list)
{
//  Τοπικές δηλώσεις
//  Εντολές
}  //  τέλος όνομα_συνάρτησης
```

όπου return_type είναι ο τύπος επιστρεφόμενης τιμής, function_name είναι το όνομα της συνάρτησης και parameter list είναι η λίστα παραμέτρων της

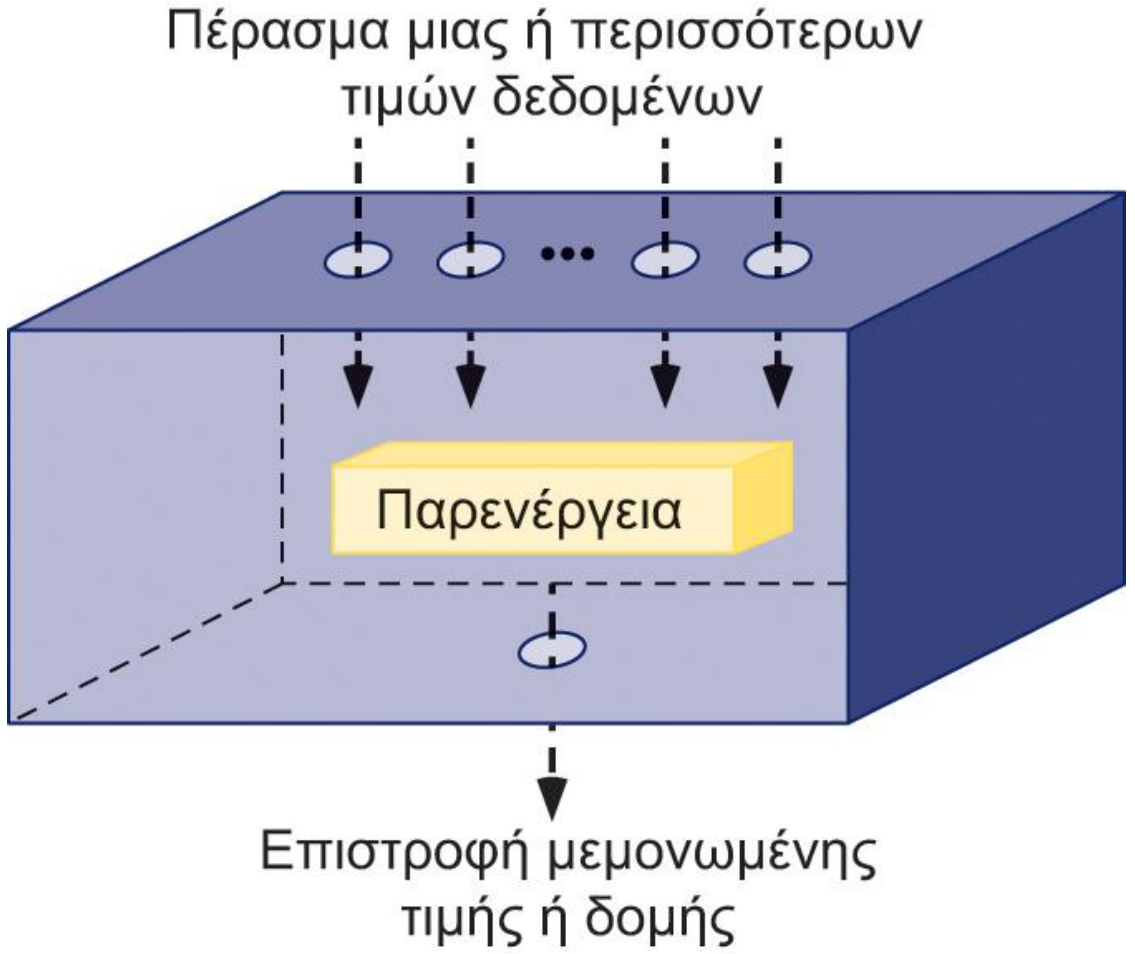
Συναρτήσεις στη C



ΣΧΗΜΑ 4-3 Διάγραμμα δομής για ένα πρόγραμμα C.

Συναρτήσεις στη C

ΣΧΗΜΑ 4-4 Σχηματική αναπαράσταση της λειτουργίας μιας συνάρτησης.



Συναρτήσεις στη C

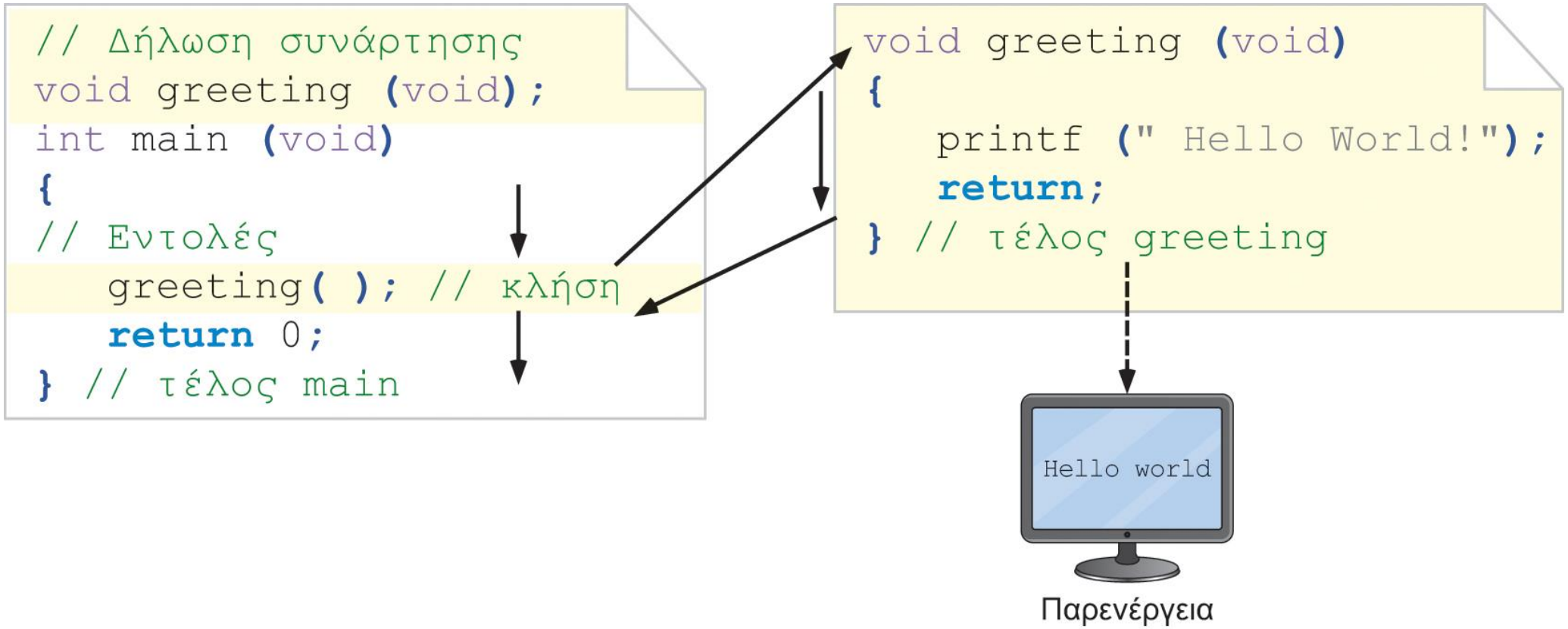
- Γιατί χρησιμοποιούμε συναρτήσεις;
  - Μάς επιτρέπουν να σ προβλήματα σε κατανοητά και εύκολα διαχειρίσιμα βήμα
  - Παρέχουν έναν τ ποίησης κώδικα
  - Οι συναρτήσεις «π δεδομένα
    - Τα τοπικά δεδομένα ορίζονται μέσα σε μία συνάρτηση- δεν είναι δυνατό να τροποποιηθούν εκτός της συνάρτησης

4.3
Συναρτήσεις καθοριζόμενες από τον χρήστη

Συναρτήσεις καθοριζόμενες από τον χρήστη

- Όπως κάθε άλλο αντικείμενο στη C, οι συναρτήσεις πρέπει να δηλώνονται και να ορίζονται
  - Η δήλωση αναφέρει το όνομα της συνάρτησης, τον τύπο της επιστρεφόμενης τιμής και τη λίστα των τυπικών παραμέτρων, δηλαδή τον τύπο και τη σειρά τους
  - Ο ορισμός της συνάρτησης, ο οποίος κατά παράδοση γράφεται μετά τη συνάρτηση που πραγματοποιεί την κλήση, περιέχει τον απαιτούμενο κώδικα για την ολοκλήρωση της εργασίας που εκτελεί η συνάρτηση

Συναρτήσεις καθοριζόμενες από τον χρήστη



ΣΧΗΜΑ 4-5 Δήλωση, κλήση και ορισμός μιας συνάρτησης με όνομα `greeting`.

Βασικές κατηγορίες συναρτήσεων

Ένα σχήμα κατηγοριοποίησης των συναρτήσεων βασίζεται στα εξής δύο στοιχεία:

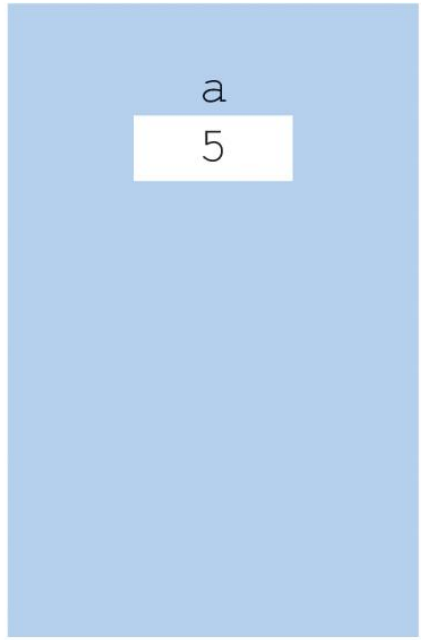
- Το εάν επιστρέφουν μία τιμή
- Το εάν δέχονται παραμέτρους ή όχι
  - Κάθε συνάρτηση είτε επιστρέφει μία τιμή, είτε όχι
  - Οι συναρτήσεις που δεν επιστρέφουν τιμή αναφέρονται ως `void`
  - Πολλές συναρτήσεις έχουν παραμέτρους, ενώ άλλες όχι

Βασικές κατηγορίες συναρτήσεων

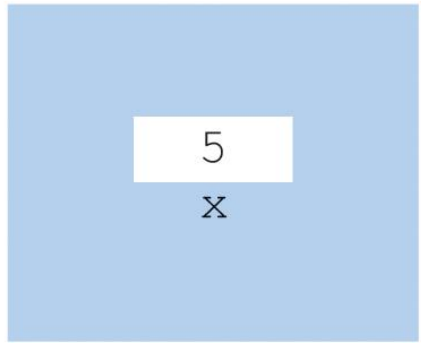
- Ο συνδυασμός των δύο παραπάνω στοιχείων οδηγεί σε τέσσερις βασικές κατηγορίες συναρτήσεων:
  - void συναρτήσεις χωρίς παραμέτρους
  - void συναρτήσεις με παραμέτρους
  - συναρτήσεις που επιστρέφουν μία τιμή αλλά δεν έχουν παραμέτρους (μη void συναρτήσεις χωρίς παραμέτρους)
  - συναρτήσεις που επιστρέφουν μία τιμή και έχουν παραμέτρους (μη void συναρτήσεις με παραμέτρους)

ΣΧΗΜΑ 4-6 Συνάρτηση **`void`** με παράμετρο.

```
// Δήλωση συνάρτησης
void printOne (int X);
int main (void )
{
// Τοπικές δηλώσεις
   int a = 5;
// Εντολές
   printOne (a); // κλήση
   return 0;
} // τέλος main
```




```
void printOne (int x)
{
   printf ("%d\n", x);
   return ;
} // τέλος printone
```



Παρενέργεια

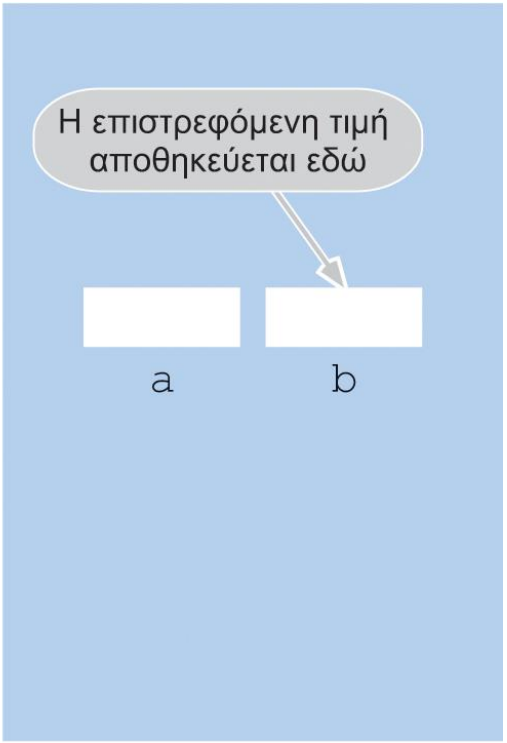
```c
// Δήλωση συνάρτησης
int getQuantity (void);
int main (void)
{
// Τοπικές δηλώσεις
   int amt;
// Εντολές
   amt = getQuantity ( );
   return 0;
} // τέλος main
```

```c
int getQuantity (void )
{
// Τοπικές δηλώσεις
   int qty;
// Εντολές
   printf ("Enter Quantity");
   scanf ("%d" , &qty);
   return qty;
} // τέλος getQuantity
```

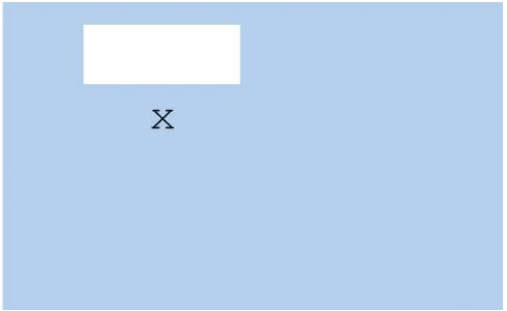
ΣΧΗΜΑ 4-7 Μη void συνάρτηση χωρίς παραμέτρους.

ΣΧΗΜΑ 4-8
Κλήση μιας συνάρτησης
η οποία επιστρέφει μία τιμή.

```
// Δήλωση συνάρτησης
int sqr ( int x);
int main (void )
{
// Τοπικές δηλώσεις
   int a;
   int b;
// Εντολές
   scanf ("%d", &a);
   b = sqr (a);
   printf ("%d squared: %d\n", a, b);
   return 0;
} // τέλος main
```




```
int sqr (int x)
{
// Εντολές
   return (x * x);
} // τέλος sqr
```



Ορισμός συνάρτησης

- Ο ορισμός μιας συνάρτησης περιέχει τον κώδικα της συνάρτησης.
- Αποτελείται από δύο μέρη: την επικεφαλίδα και το σώμα της συνάρτησης
- Η επικεφαλίδα συνάρτησης (function header) αποτελείται από τρία μέρη: τον τύπο επιστρεφόμενης τιμής, το όνομα της συνάρτησης και την λίστα τυπικών παραμέτρων
- Το σώμα συνάρτησης (function body) περιέχει τις τοπικές δηλώσεις και τις εντολές της συνάρτησης
  - Το σώμα ξεκινά με τους τοπικούς ορισμούς που καθορίζουν τις μεταβλητές που χρειάζεται η συνάρτηση
  - Μετά τους τοπικούς ορισμούς, γράφονται οι εντολές της συνάρτησης, οι οποίες καταλήγουν σε μία εντολή **`return`**

Ορισμός συνάρτησης

Κεφαλίδα Συνάρτησης

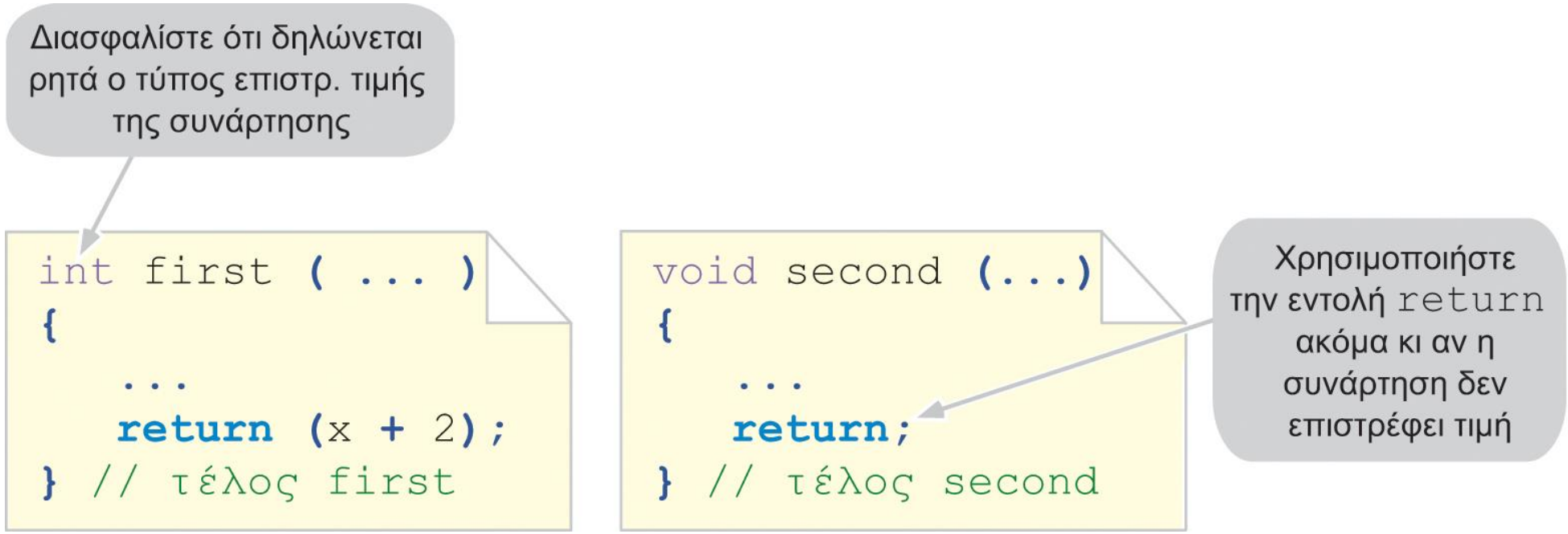
```
τύπος_επιστρ. τιμής όνομα_συνάρτησης (λίστα τυπικών παραμέτρων)
```

```
{
// Τοπικές δηλώσεις
   …
// Εντολές
   …
} // τέλος όνομα_συνάρτησης
```

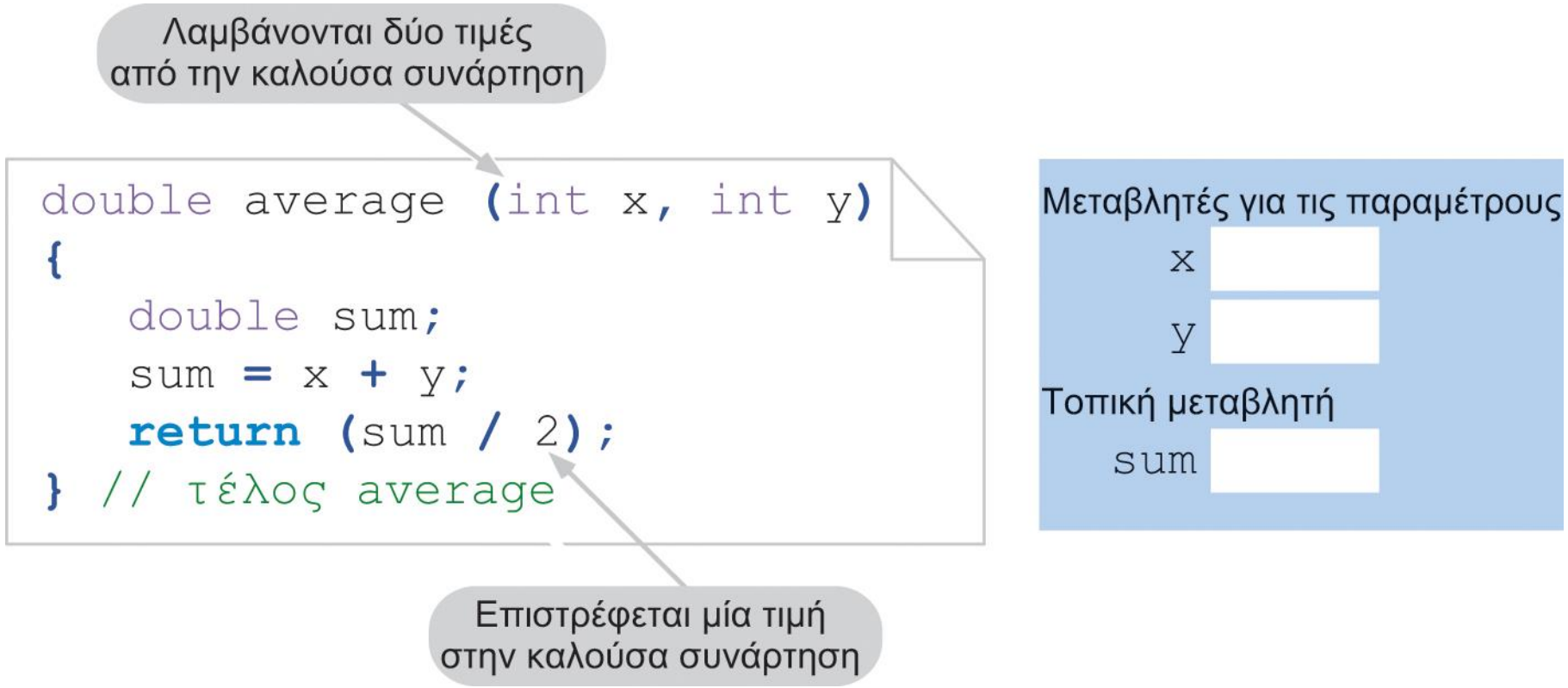
Σώμα συνάρτησης

Ορισμός συνάρτησης

- Λίστα τυπικών παραμέτρων
  - Στον ορισμό μιας συνάρτησης, οι παράμετροι περιέχονται στην λίστα τυπικών παραμέτρων
    - Αυτή η λίστα ορίζει και δηλώνει τις μεταβλητές που θα δεχτούν τα δεδομένα που λαμβάνει η συνάρτηση
- Τοπικές μεταβλητές
  - Μία τοπική μεταβλητή είναι μία μεταβλητή που ορίζεται μέσα σε μία συνάρτηση και χρησιμοποιείται εντός αυτής, χωρίς να έχει κανένα ρόλο στην επικοινωνία μεταξύ των συναρτήσεων



ΣΧΗΜΑ 4-10 Ο ρόλος της εντολής `return` σε συναρτήσεις



ΣΧΗΜΑ 4-11 Τοπικές μεταβλητές συνάρτησης.

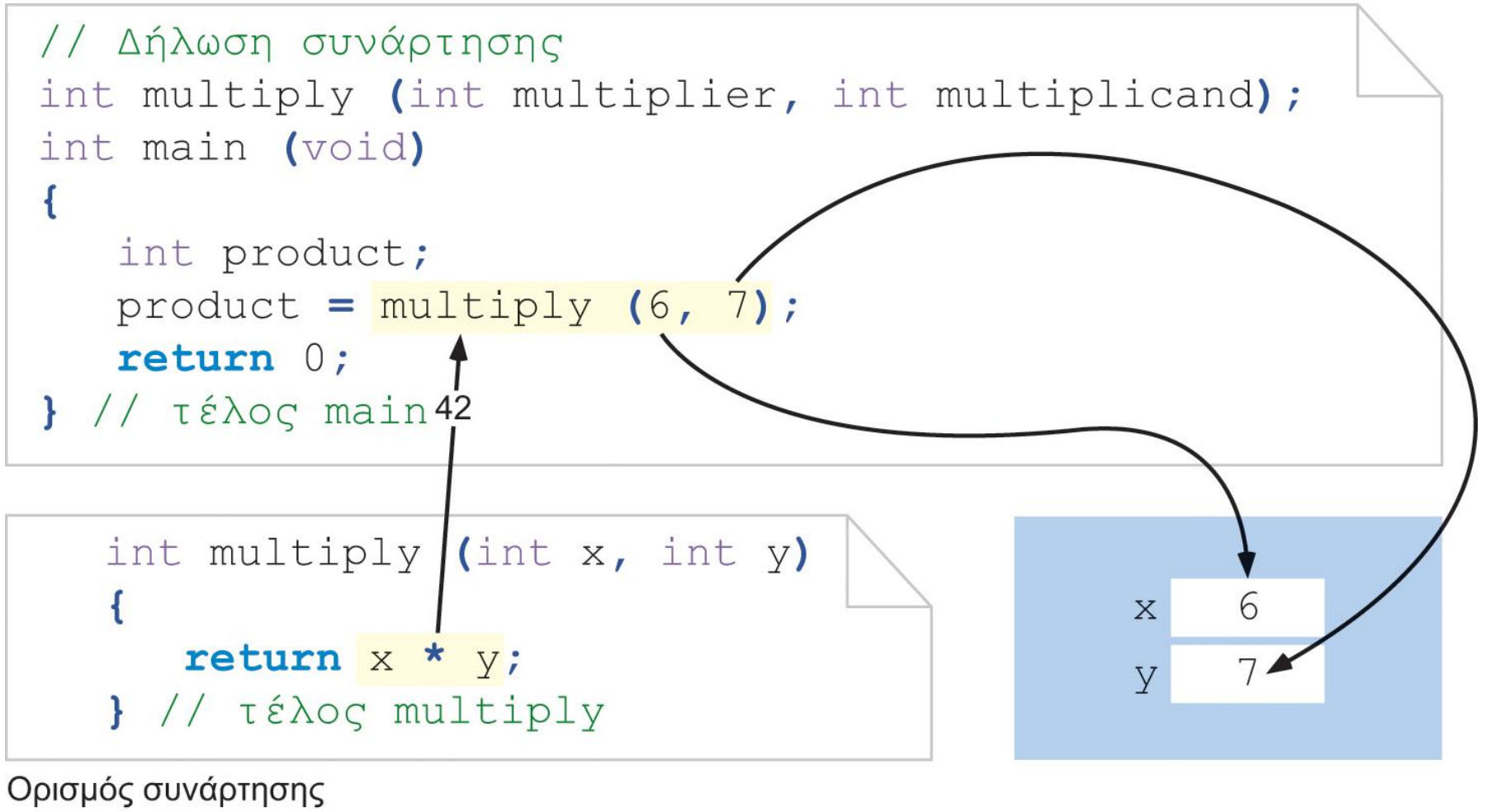
Δήλωση συνάρτησης

- Η δήλωση συνάρτησης αποτελείται μόνο από την επικεφαλίδα της συνάρτησης—δεν περιέχει κώδικα
- Οι δηλώσεις συναρτήσεων αποτελούνται από τρία μέρη:
  - τον τύπο επιστρεφόμενης τιμής
  - το όνομα της συνάρτησης
  - τη λίστα τυπικών παραμέτρων
- Οι δηλώσεις συναρτήσεων τερματίζονται με χαρακτήρα ελληνικού ερωτηματικού

Κλήση συνάρτησης

- Η κλήση συνάρτησης (function call) είναι μία επιθεματική έκφραση με προτεραιότητα 16
  - Ο τελεστέος σε μία κλήση συνάρτησης είναι το όνομα της συνάρτησης
  - Ο τελεστής είναι το ζεύγος παρενθέσεων, (...), το οποίο περιέχει τις πραγματικές παραμέτρους της συνάρτησης
- Οι πραγματικές παράμετροι προσδιορίζουν τις τιμές που πρέπει να αποσταλούν στην καλούμενη συνάρτηση
  - Αντιστοιχούν στις τυπικές παραμέτρους που αναφέρονται στη δήλωση της συνάρτησης, τόσο ως προς τον τύπο όσο και ως προς τη θέση τους στη λίστα παραμέτρων

Κλήση συνάρτησης



ΣΧΗΜΑ 4-12 Η «ανατομία» μιας κλήσης συνάρτησης.

Κλήση συνάρτησης

```
multiply (6, 7)                    multiply (a, 7)
multiply (6, b)                    multiply (a + 6, 7)
multiply (multiply (a, b), 7)      multiply (... , ...)
```

ΣΧΗΜΑ 4-12 Παραδείγματα κλήσεων συναρτήσεων.

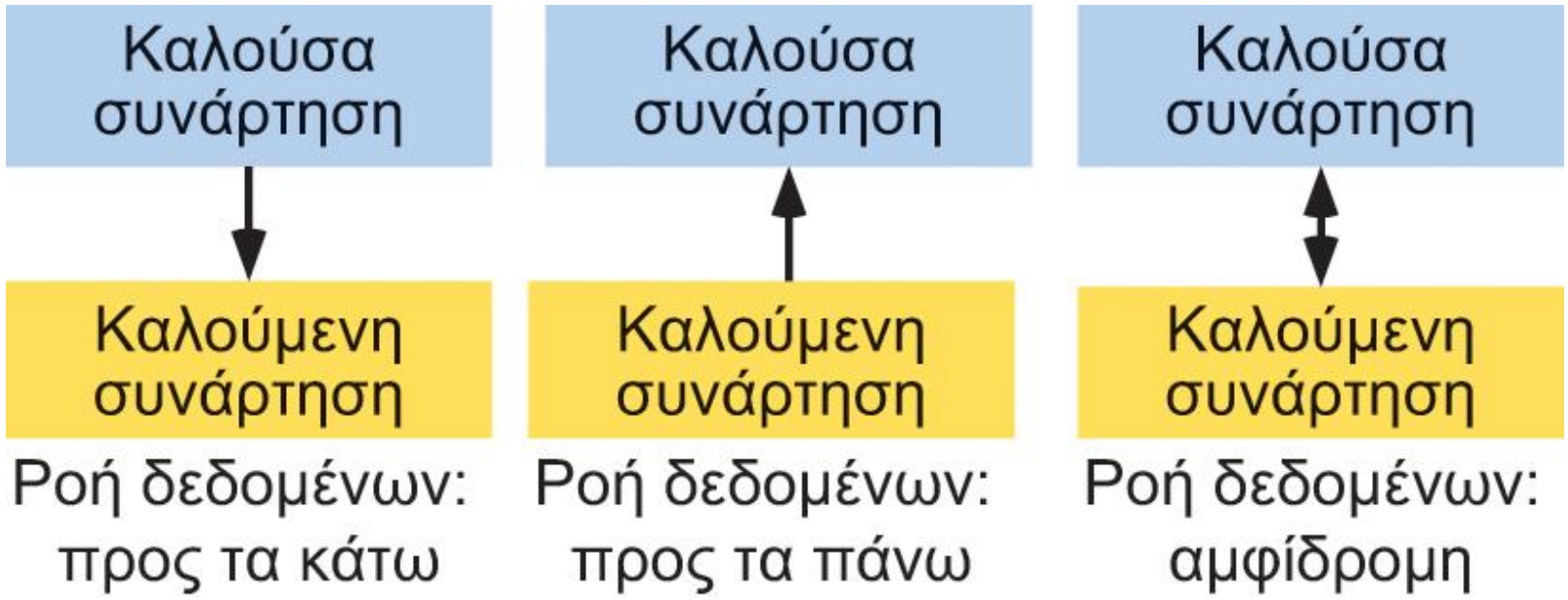
4.4
Επικοινωνία μεταξύ συναρτήσεων

Επικοινωνία μεταξύ συναρτήσεων

- Η καλούσα και η καλούμενη συνάρτηση είναι δύο ξεχωριστές οντότητες, αλλά πρέπει να επικοινωνούν για να ανταλλάσσουν δεδομένα
- Η ροή δεδομένων μεταξύ της καλούσας και της καλούμενης συνάρτησης μπορεί να έχει τρεις διαφορετικές μορφές:
  - Ροή προς τα κάτω, από την καλούσα προς την καλούμενη συνάρτηση
  - Ροή προς τα επάνω, από την καλούμενη προς την καλούσα συνάρτηση
  - Αμφίδρομη ροή δεδομένων και προς τις δύο κατευθύνσεις

Ο όρος ροή αναφέρεται στο πέρασμα δεδομένων

Επικοινωνία μεταξύ συναρτήσεων



ΣΧΗΜΑ 4-16 Τρεις μορφές ροής δεδομένων.

Βασική αρχή επικοινωνίας μεταξύ συναρτήσεων

- Ροή προς τα κάτω: η καλούσα συνάρτηση στέλνει (περνάει) δεδομένα στην καλούμενη συνάρτηση
  - Δεν υπάρχει ροή δεδομένων προς την αντίθετη κατεύθυνση
- Ροή προς τα επάνω: η καλούμενη συνάρτηση στέλνει δεδομένα πίσω στην καλούσα συνάρτηση χωρίς να λαμβάνει δεδομένα από αυτήν
- Αμφίδρομη ροή δεδομένων:
  - Η καλούσα συνάρτηση στέλνει δεδομένα προς τα κάτω, στην καλούμενη συνάρτηση
  - Κατά τη διάρκεια ή στο τέλος της επεξεργασίας της, η καλούμενη συνάρτηση στέλνει δεδομένα προς τα επάνω, στην καλούσα συνάρτηση

Υλοποίηση στη C

- Ροή δεδομένων μπορεί να υπάρχει
  - από την καλούσα στην καλούμενη συνάρτηση, ή
  - από την καλούμενη συνάρτηση πίσω στην καλούσα
- Επικοινωνία προς τα κάτω
  - Μονόδρομη μορφή επικοινωνίας: η καλούσα συνάρτηση μπορεί να στείλει δεδομένα στην καλούμενη συνάρτηση, αλλά η καλούμενη συνάρτηση δεν μπορεί να στείλει δεδομένα στην καλούσα συνάρτηση

Υλοποίηση στη C

- Επικοινωνία προς τα επάνω
  - Η C παρέχει μόνο έναν μηχανισμό για την «οδήγηση» της ροής εκτέλεσης προς τα επάνω: τη εντολή **`return`**
- Αμφίδρομη επικοινωνία
  - Ροή δεδομένων και προς τις δύο κατευθύνσεις

4.5
Πρότυπες συναρτήσεις

Πρότυπες συναρτήσεις

- Η C παρέχει μία πλούσια συλλογή από πρότυπες συναρτήσεις (standard functions)
- Οι ορισμοί αυτών των συναρτήσεων περιλαμβάνονται στις πρότυπες βιβλιοθήκες της γλώσσας και μπορούν να χρησιμοποιούνται σε οποιοδήποτε πρόγραμμα
  - Για να χρησιμοποιηθεί μία πρότυπη συνάρτηση σε ένα πρόγραμμα, πρέπει να συμπεριληφθεί σε αυτό η δήλωσή της
  - Οι δηλώσεις των πρότυπων συναρτήσεων είναι συγκεντρωμένες σε αρχεία κεφαλίδας (header files)

Πρότυπες συναρτήσεις

```c
#include <stdio.h>

int main (void )
{
    ...
    scanf ( ... );
    ...
    printf ( ... );
    ...
    return 0;
} // τέλος main
```

Παράγεται από το αρχείο κεφαλίδας `stdio`

```c
int scanf( ... );
int printf( ... );
```

Το πρόγραμμά μας

```c
int main (void)
{
    ...
    scanf ( ... );
    ...
    printf ( ... );
    ...
    return 0;
} // τέλος main
```

Αντί να προσθέτουμε τη δήλωση για κάθε μεμονωμένη συνάρτηση που χρειαζόμαστε, συμπεριλαμβάνουμε απλώς τα κατάλληλα αρχεία κεφαλίδας στην αρχή του προγράμματος

Ορισμοί που προστίθενται από τον συνδέτη

```c
int scanf ( ... )
{
    ...
    return ...;
} // τέλος scanf
```

```c
int printf( ... )
{
    ...
    return ...;
} // τέλος printf
```

Μαθηματικές συναρτήσεις

- Συναρτήσεις απόλυτης τιμής
  - Απόλυτη τιμή: αναφέρεται στο (θετικό) μέγεθος ή μέτρο μιας τιμής, ανεξάρτητα από το πρόσημό της
- Συναρτήσεις χειρισμού μιγαδικών αριθμών
  - Οι συναρτήσεις που διαθέτει η C για το χειρισμό μιγαδικών αριθμών είναι συγκεντρωμένες στο αρχείο κεφαλίδας `complex.h`
- Συναρτήσεις στρογγυλοποίησης προς τα επάνω
  - Άνω όριο: Η μικρότερη ακέραια τιμή που είναι μεγαλύτερη ή ίση με έναν αριθμό
- Συναρτήσεις στρογγυλοποίησης προς τα κάτω
  - Κάτω όριο: Η μεγαλύτερη ακέραια τιμή που είναι μικρότερη ή ίση με έναν αριθμό

Μαθηματικές συναρτήσεις

- Συναρτήσεις αποκοπής
  - Επιστρέφουν το ακέραιο μέρος ενός αριθμού, εκτελώντας στρογγυλοποίηση προς τα κάτω (προς την κατεύθυνση του μηδενός).
- Συναρτήσεις στρογγυλοποίησης
  - Επιστρέφουν την πλησιέστερη ακέραια τιμή
- Συναρτήσεις ύψωσης σε δύναμη
  - Επιστρέφουν την τιμή του πρώτου ορίσματος υψωμένη στη δύναμη του δεύτερου, δηλαδή, το αποτέλεσμα της πράξης n1n2
- Συναρτήσεις τετραγωνικής ρίζας
  - Επιστρέφουν την τετραγωνική ρίζα του αριθμού που λαμβάνουν ως όρισμα (μία μη αρνητική τιμή)

Τυχαίοι αριθμοί

- Η C παρέχει δύο συναρτήσεις για την παραγωγή μιας ακολουθίας τυχαίων αριθμών
  - **`srand`**: παραγωγή τυχαίων αριθμών με τυχαία τιμή σποράς
  - **`rand`**: για παραγωγή τυχαίων αριθμών με δεδομένη τιμή σποράς
- Χρήση τυχαίων αριθμών
  - Η συνάρτηση **`rand`** μπορεί να επιστρέφει αριθμούς σε μεγάλο εύρος τιμών
  - Σύμφωνα με το πρότυπο της C, αυτό το εύρος τιμών είναι τουλάχιστον από 0 έως 32.767

4.6
Εμβέλεια

Εμβέλεια

- Ο όρος εμβέλεια (scope) αναφέρεται στην ορατότητα ενός στοιχείου από διάφορα σημεία του προγράμματος
  - Υποδεικνύει την περιοχή του προγράμματος στην οποία μπορούμε να χρησιμοποιήσουμε το όνομα του στοιχείου
- Εμβέλεια έχει κάθε στοιχείο το οποίο μπορεί να δηλωθεί σε ένα πρόγραμμα, π.χ., μία μεταβλητή ή μία δήλωση συνάρτησης
  - Δεν ισχύει για τις οδηγίες προς τον προμεταγλωττιστή, όπως οι εντολές define (διέπονται από ξεχωριστούς κανόνες)
- Η έννοια της εμβέλειας αφορά τον πηγαίο κώδικα
  - Δεν υφίσταται στο εκτελέσιμο πρόγραμμα

Γενική (καθολική) εμβέλεια

- Οποιοδήποτε στοιχείο ορίζεται στην περιοχή γενικής εμβέλειας, είναι ορατό από το σημείο ορισμού του μέχρι το τέλος του προγράμματος
  - λέγεται ότι έχει καθολική εμβέλεια

ΣΧΗΜΑ 4-32 Η έννοια της εμβέλειας και τα διάφορα επίπεδα εμβέλειας.

```
/* Παράδειγμα, ΜΟΝΟ για σκοπούς επίδειξης. Ο τρόπος γραφής αυτού του
   προγράμματος δεν θα πρέπει να χρησιμοποιείται ποτέ στην πράξη
*/
#include <stdio.h>
int fun (int a, int b);
```

Καθολική εμβέλεια

```
int main (void )
{
   int a;
   int b;
   float y;
   ...

   {
     // Αρχή ένθετου μπλοκ
        float a = y / 2;
        float y;
        float z;
        ...
        z = a* b;
        ...
   } // Τέλος ένθετου μπλοκ

   ...
} // τέλος main
```

Εμβέλεια της `main`

Εμβέλεια μπλοκ

```
int fun (int i, int j)
{
   int a;
   int y;
   ...
} // τέλος fun
```

Εμβέλεια της `fun`

Τοπική εμβέλεια

- Οι μεταβλητές που ορίζονται μέσα σε ένα μπλοκ κώδικα έχουν τοπική εμβέλεια (local scope)
  - Η ύπαρξη και, κατ' επέκταση, η ορατότητά τους περιορίζεται από το σημείο δήλωσής τους μέχρι το τέλος του μπλοκ (συνήθως μία συνάρτηση) στο οποίο δηλώνονται
  - Έξω από αυτό το μπλοκ, είναι αόρατες

Σύνοψη

- Σύμφωνα με τις αρχές του δομημένου προγραμματισμού, ένα πρόγραμμα χωρίζεται σε μονάδες.
  - Κάθε μονάδα σχεδιάζεται κατά τρόπο ώστε να εκτελεί μία συγκεκριμένη εργασία.
  - Στη C, οι μονάδες γράφονται ως συναρτήσεις.
- Κάθε πρόγραμμα C πρέπει να έχει μία και μόνο μία συνάρτηση με όνομα main.
- Μία συνάρτηση μπορεί να επιστρέφει μόνο μία τιμή.
- Μία συνάρτηση μπορεί να καλείται για την επιστρεφόμενη τιμή της, ή για την παρενέργειά της (το «δευτερεύον αποτέλεσμα» της).

Σύνοψη

- Η κλήση συνάρτησης περιλαμβάνει το όνομα της συνάρτησης και τις τιμές των παραμέτρων, δηλαδή τα πραγματικά δεδομένα που χρειάζεται η καλούμενη συνάρτηση για να εκτελέσει την εργασία της.

- Κάθε πραγματική παράμετρος της συνάρτησης είναι μία έκφραση.
Η έκφραση πρέπει να μπορεί να αποτιμηθεί σε μία τιμή τη στιγμή που καλείται η συνάρτηση.

- Εάν μία συνάρτηση δεν επιστρέφει τιμή, ο τύπος επιστρεφόμενης τιμής πρέπει να δηλωθεί ως void.

- Εάν μία συνάρτηση δεν έχει παραμέτρους, η λίστα παραμέτρων πρέπει να δηλώνεται κενή.

Σύνοψη

- Οι πραγματικές παράμετροι που περνιούνται σε μία συνάρτηση πρέπει να συμφωνούν ως προς τον αριθμό, τον τύπο και τη σειρά, με τις τυπικές παραμέτρους που αναφέρονται στον ορισμό της συνάρτησης.

- Μία δήλωση συνάρτησης απαιτεί τα εξής: τον τύπο επιστρεφόμενης τιμής και το όνομα της συνάρτησης και τον αριθμό, τους τύπους και τη σειρά των τυπικών παραμέτρων. Μπορούν να προστεθούν αναγνωριστικά για τις παραμέτρους, για λόγους τεκμηρίωσης, αλλά δεν είναι υποχρεωτικά.

- Όταν καλείται μία συνάρτηση, ο έλεγχος μεταβιβάζεται στην καλούμενη συνάρτηση. Η καλούσα συνάρτηση τελεί «σε αναμονή» μέχρι η καλούμενη συνάρτηση να ολοκληρώσει την εργασία της.

Σύνοψη

- Αν και δεν είναι υποχρεωτικό να υπάρχει εντολή `return` σε κάθε συνάρτηση, η χρήση της αποτελεί καλή πρακτική. Η `return` απαιτείται εάν ο τύπος επιστρεφόμενης τιμής είναι οτιδήποτε άλλο εκτός από `void`.

- Ο έλεγχος επιστρέφει στον κώδικα που έκανε την κλήση (στην καλούσα συνάρτηση) όταν η ροή εκτέλεσης συναντά μία εντολή `return`.

- Η εμβέλεια μιας παραμέτρου εκτείνεται στο τμήμα (μπλοκ) κώδικα που ακολουθεί μετά την επικεφαλίδα της συνάρτησης.

- Μία τοπική μεταβλητή υφίσταται (έχει εμβέλεια) μόνο μέσα στη συνάρτηση στην οποία ορίζεται. Οι τοπικές μεταβλητές δεν συμμετέχουν στην επικοινωνία μεταξύ της καλούσας και της καλούμενης συνάρτησης.

Σύνοψη

- Τα διαγράμματα δομής χρησιμοποιούνται για τη σχεδίαση προγραμμάτων. Δεν περιέχουν κώδικα.

- Η λειτουργική συνεκτικότητα αποτελεί ένα μέτρο του πόσο στενά σχετίζονται οι διεργασίες που εκτελούνται εντός μιας συνάρτησης.

- Η διαδικασία υλοποίησης προγραμμάτων πρέπει να ακολουθεί τη σχεδίαση «από πάνω προς τα κάτω».

- Για τις συναρτήσεις που δεν έχουν γραφεί σε ένα συγκεκριμένο στάδιο της υλοποίησης, μπορούν να χρησιμοποιούνται κενές συναρτήσεις προσομοίωσης (stubs).