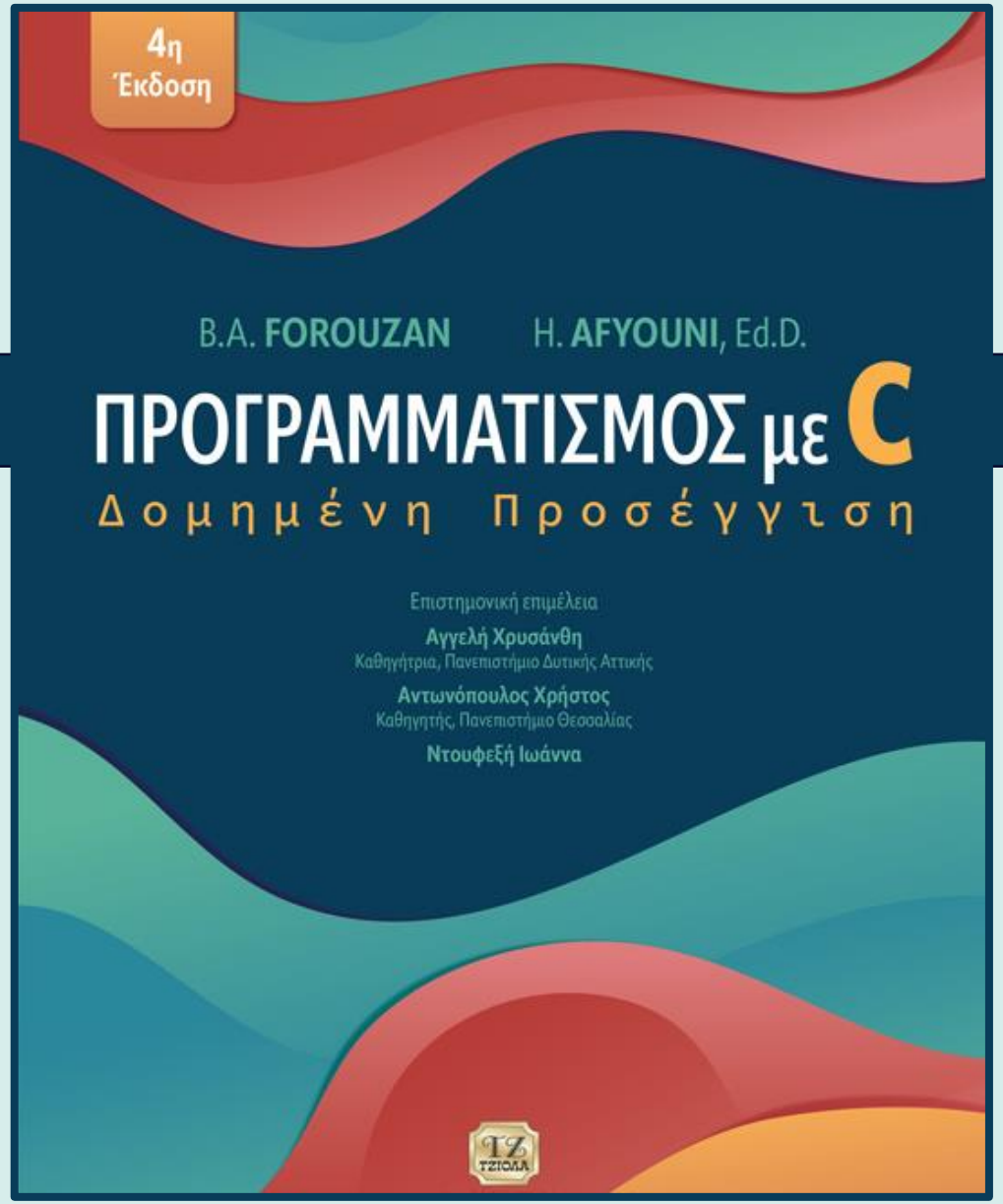


Επιστημονικές Εκδόσεις
ΤΖΙΟΛΑ

ΚΕΦΑΛΑΙΟ 5

Δομές Επιλογής και Λήψης Αποφάσεων

ΘΕΜΑΤΑ ΚΕΦΑΛΑΙΟΥ

- Χρήση των λογικών τελεστών and, or και not για τη λήψη αποφάσεων
- Υλοποίηση επιλογών δύο κατευθύνσεων σε προγράμματα
- Υλοποίηση επιλογών πολλαπλών κατευθύνσεων σε προγράμματα
- Χρήση λογικών τελεστών και τελεστών σύγκρισης σε προγράμματα

5.1
Λογικά δεδομένα και λογικοί τελεστές

Λογικά δεδομένα και λογικοί τελεστές

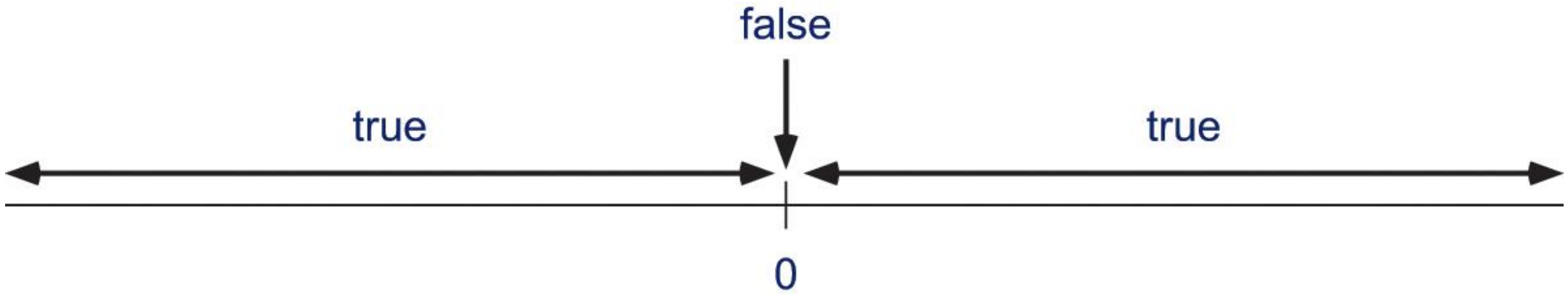
- Μία εντολή επιλογής μάς δίνει τη δυνατότητα να επιλέξουμε μεταξύ δύο ή περισσότερων εναλλακτικών «περιπτώσεων»
  - Με άλλα λόγια, μας επιτρέπει να λαμβάνουμε αποφάσεις
- Οι αποφάσεις που καλείται να λάβει ένας υπολογιστής πρέπει να είναι πολύ απλές, επειδή τα πάντα στον υπολογιστή ανάγονται, τελικά, σε κάτι το οποίο αποτιμάται ως true (αληθές) ή false (ψευδές)
  - Σε πολυπλοκότερες περιπτώσεις λήψης αποφάσεων, είναι ευθύνη του προγραμματιστή να τις αναγάγει σε μία αλληλουχία απλών αποφάσεων που μπορεί να χειριστεί ο υπολογιστής
  - Ένα στοιχείο δεδομένων χαρακτηρίζεται ως «λογικό» εάν μεταφέρει την ιδέα του αληθούς ή του ψευδούς

Λογικά δεδομένα στη C

- Στις απαρχές της, η γλώσσα C δεν διέθετε ειδικό τύπο για την αναπαράσταση λογικών δεδομένων
- Οι προγραμματιστές της C χρησιμοποιούσαν άλλους τύπους, όπως ο int, για την αναπαράσταση λογικών δεδομένων
  - Εάν ένα στοιχείο δεδομένων έχει τιμή μηδέν, θεωρείται ότι αντιπροσωπεύει τη λογική τιμή false-ψευδές
  - Εάν ένα στοιχείο δεδομένων έχει τιμή διάφορη του μηδενός, θεωρείται ότι αντιπροσωπεύει τη λογική τιμή true-αληθές

Λογικά δεδομένα στη C

- Το πρότυπο C99 εισήγαγε έναν ειδικό τύπο για λογικά (Boolean) δεδομένα
  - Το ακριβές όνομα είναι _bool
  - Δηλώνεται ως μη προσημασμένος ακέραιος στο αρχείο κεφαλίδας stdbool.h
  - Σε αυτό το αρχείο κεφαλίδας ορίζονται επίσης τα αναγνωριστικά bool (το όνομα του τύπου), true και false (οι τιμές του τύπου)



ΣΧΗΜΑ 5-1 True και false—αληθές και ψευδές—σε αριθμητική κλίμακα.

Λογικοί τελεστές (1/3)

- Η C διαθέτει τρεις λογικούς τελεστές (logical operators) για τον συνδυασμό λογικών τιμών και τη δημιουργία νέων λογικών τιμών: *not*, *and* και *or*

not (άρνηση)

x	!x
false	true
true	false

true: αληθές
false: ψευδές

and (σύζευξη)

x	y	x && y
false	false	false
false	true	false
true	true	false
true	true	true

or (διάζευξη)

x	y	x || y
false	false	false
false	true	true
true	false	true
true	true	true

ΣΧΗΜΑ 5-2 Πίνακες αλήθειας των λογικών τελεστών.

Λογικοί τελεστές (2/3)

- Ο τελεστής λογικής άρνησης, not (!)
  - Μοναδιαίος τελεστής με προτεραιότητα 15
  - Εκτελεί λογική άρνηση του τελεστέου του—δηλαδή αλλάζει μία τιμή true σε false και το αντίστροφο
- Ο τελεστής λογικής σύζευξης, and (&)
  - Δυαδικός τελεστής με προτεραιότητα 5
  - Υπάρχουν τέσσερις πιθανοί συνδυασμοί τιμών για τους τελεστέους του
  - Εκτελεί λογική σύζευξη: το αποτέλεσμα είναι true μόνο όταν αμφότεροι οι τελεστέοι είναι true—σε όλες τις άλλες περιπτώσεις, είναι false

Λογικοί τελεστές (3/3)

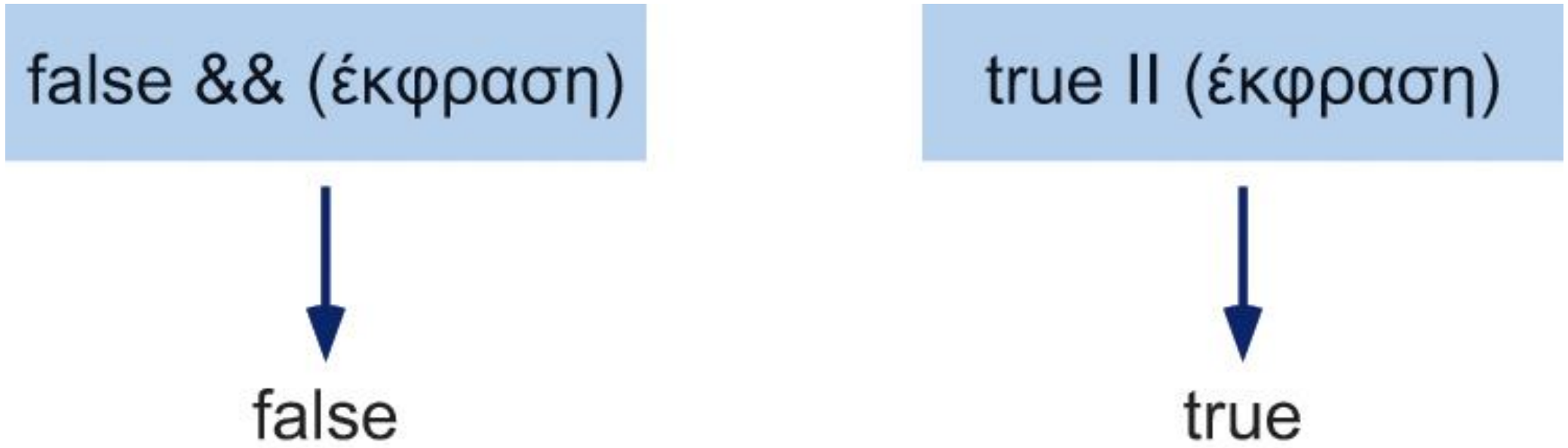
- Ο τελεστής λογικής διάζευξης, or (||)
  - δυαδικός τελεστής με προτεραιότητα 4
  - Υπάρχουν τέσσερις πιθανοί συνδυασμοί τιμών για τους τελεστέους του
  - Ο τελεστής *or* εκτελεί λογική διάζευξη: το αποτέλεσμα είναι false εάν αμφότεροι οι τελεστέοι είναι false· σε όλες τις άλλες περιπτώσεις είναι true

Αποτίμηση λογικών εκφράσεων (1/2)

- Οι γλώσσες υπολογιστών υποστηρίζουν δύο μεθόδους για την αποτίμηση λογικών εκφράσεων, δηλαδή εκφράσεων που χρησιμοποιούν δυαδικούς λογικούς τελεστές για τη διατύπωση λογικών σχέσεων
  - Η πρώτη μέθοδος απαιτεί την πλήρη αποτίμηση της έκφρασης για να καταστεί δυνατός ο προσδιορισμός του αποτελέσματος
    - μία λογική έκφραση σύζευξης (*and*) πρέπει να αποτιμηθεί πλήρως, ακόμα κι όταν ο πρώτος τελεστέος της έχει τιμή false και άρα είναι από νωρίς γνωστό ότι το αποτέλεσμα είναι υποχρεωτικά false
    - σε μία λογική έκφραση διάζευξης (*or*) πρέπει να αποτιμηθεί ολόκληρη η έκφραση ακόμα κι όταν ο πρώτος τελεστέος είναι true, οπότε το αποτέλεσμα της έκφρασης είναι υποχρεωτικά true

Αποτίμηση λογικών εκφράσεων (2/2)

- Η δεύτερη μέθοδος παρέχει το αποτέλεσμα αμέσως μόλις γίνει προσδιορίσιμο βάσει των κανόνων της άλγεβρας Boole
  - Δεν απαιτεί να ολοκληρωθεί η αποτίμηση της έκφρασης, λειτουργεί ως μηχανισμός συντόμευσης που σταματά την αποτίμηση αφ' ης στιγμής είναι γνωστό με βεβαιότητα το τελικό αποτέλεσμα
  - Εάν ο πρώτος τελεστέος μιας λογικής έκφρασης σύζευξης έχει τιμή false, το δεύτερο μισό της έκφρασης δεν αποτιμάται, επειδή είναι προφανές ότι το αποτέλεσμα είναι κατ' ανάγκην false
  - Το ίδιο ισχύει και για τις λογικές εκφράσεις διάζευξης: εάν ο πρώτος τελεστέος έχει τιμή true, δεν είναι απαραίτητο να αποτιμηθεί το δεύτερο μισό της έκφρασης και το αποτέλεσμα τίθεται αμέσως σε true



ΣΧΗΜΑ 5-3 Συντόμευση της αποτίμησης λογικών εκφράσεων and/or.

Τελεστές σύγκρισης (1/3)

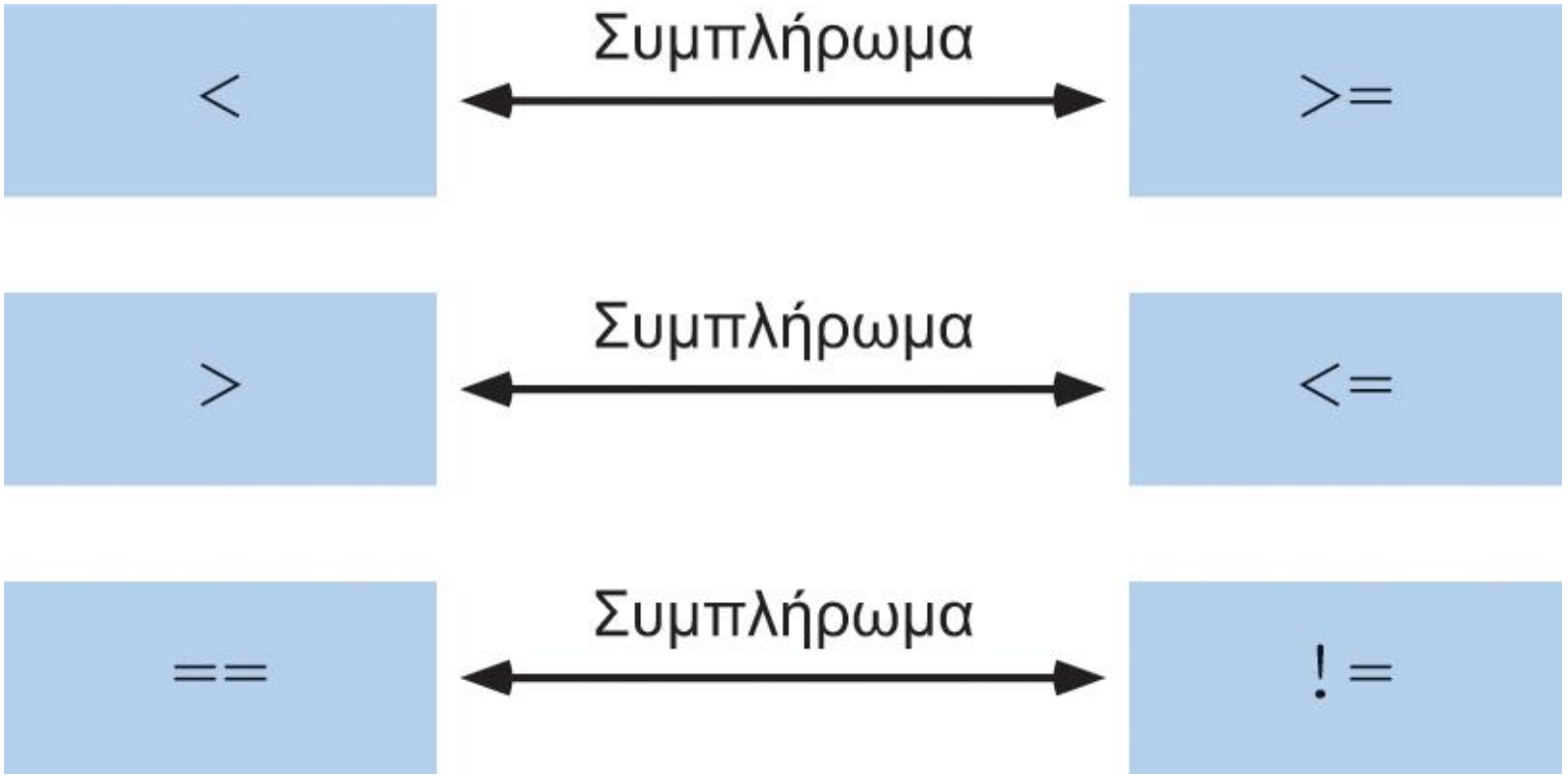
- Επιπρόσθετα με τους λογικούς τελεστές, η C υποστηρίζει έξι τελεστές σύγκρισης
- Οι τελεστές σύγκρισης χωρίζονται σε δύο κατηγορίες, τους σχεσιακούς και τους τελεστές ισότητας
  - Όλοι οι τελεστές σύγκρισης είναι δυαδικοί: καλούνται να συγκρίνουν δύο τελεστέους
  - Το αποτέλεσμα είναι πάντα τύπου Boolean, δηλαδή μία τιμή true ή false

Τελεστές σύγκρισης (2/3)

ΠΙΝΑΚΑΣ 5-2 Τελεστές σύγκρισης: σχεσιακοί και ισότητας

ΚΑΤΗΓΟΡΙΑ	ΤΕΛΕΣΤΗΣ	ΣΗΜΑΣΙΑ	ΠΡΟΤΕΡΑΙΟΤΗΤΑ
Σχεσιακοί	< <= > >=	μικρότερο από μικρότερο από ή ίσο με μεγαλύτερο από μεγαλύτερο από ή ίσο με	10
Ισότητας	== !=	ίσο άνισο	9

Τελεστές σύγκρισης (3/3)



ΣΧΗΜΑ 5-4 Τελεστές σύγκρισης και το συμπλήρωμά τους.

ΠΙΝΑΚΑΣ 5-3 Απλοποίηση εκφράσεων με χρήση συμπληρωματικών τελεστών

ΑΡΧΙΚΗ ΕΚΦΡΑΣΗ	ΑΠΛΟΠΟΙΗΜΕΝΗ ΕΚΦΡΑΣΗ
`!(x < y)`	`x >= y`
`!(x > y)`	`x <= y`
`!(x != y)`	`x == y`
`!(x <= y)`	`x > y`
`!(x >= y)`	`x < y`
`!(x == y)`	`x != y`

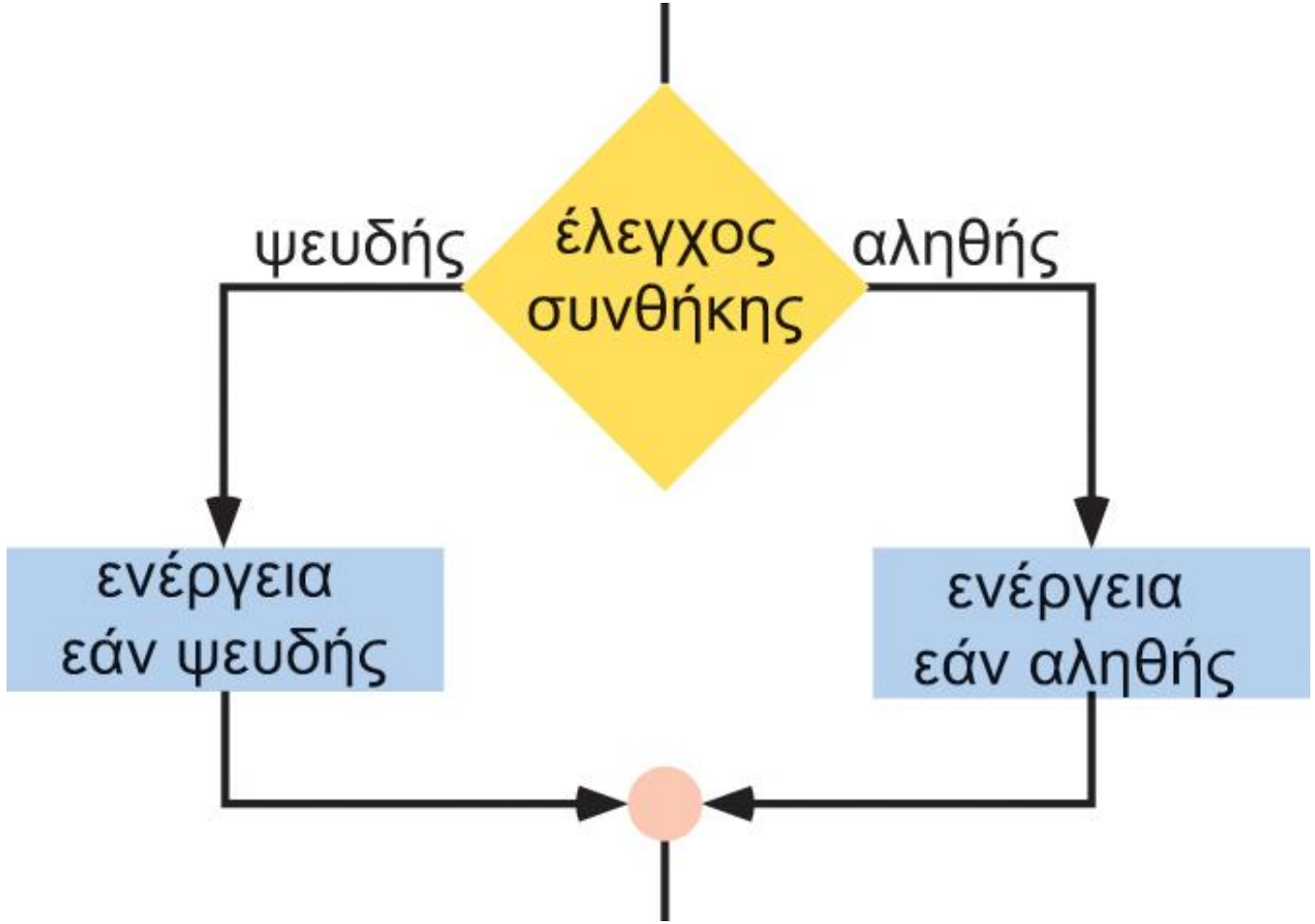
5.2
Επιλογή δύο κατευθύνσεων

Επιλογή δύο κατευθύνσεων (1/2)

- Εντολή επιλογής δύο κατευθύνσεων: επιτρέπει λήψη αποφάσεων όταν υπάρχουν δύο εναλλακτικές επιλογές
- Ο υπολογιστής την επεξεργάζεται ως εντολή εκτέλεσης υπό συνθήκη, όπου η «συνθήκη» μπορεί να αποτιμηθεί είτε ως true-αληθής, είτε ως false-ψευδής
    - Εάν η συνθήκη αποτιμάται ως true, εκτελείται μία ενέργεια ή ένα σύνολο ενεργειών
    - Εάν η συνθήκη αποτιμάται ως false, εκτελείται μία διαφορετική ενέργεια ή σύνολο ενεργειών
    - Ανεξάρτητα από το ποιο από τα δύο σύνολα ενεργειών εκτελείται για την εντολή επιλογής, η ροή εκτέλεσης του προγράμματος συνεχίζει με την εντολή που ακολουθεί μετά από την εντολή επιλογής

Επιλογή δύο κατευθύνσεων (2/2)

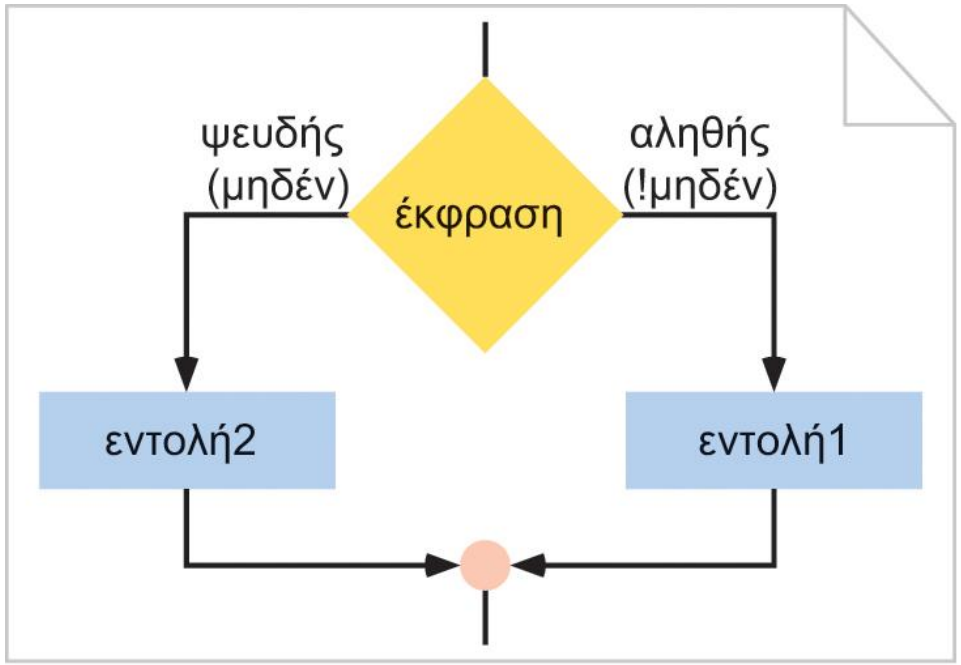
ΣΧΗΜΑ 5-5 Λογική ροή της διαδικασίας επιλογής δύο κατευθύνσεων.



Η εντολή `if ... else`

- Η C υλοποιεί την επιλογή δύο κατευθύνσεων με τη εντολή if ... else
  - Η if ... else είναι μία σύνθετη εντολή η οποία χρησιμοποιείται για τη λήψη μιας απόφασης μεταξύ δύο εναλλακτικών επιλογών

ΣΧΗΜΑ 5-6 Λογική ροή της εντολής `if … else`.

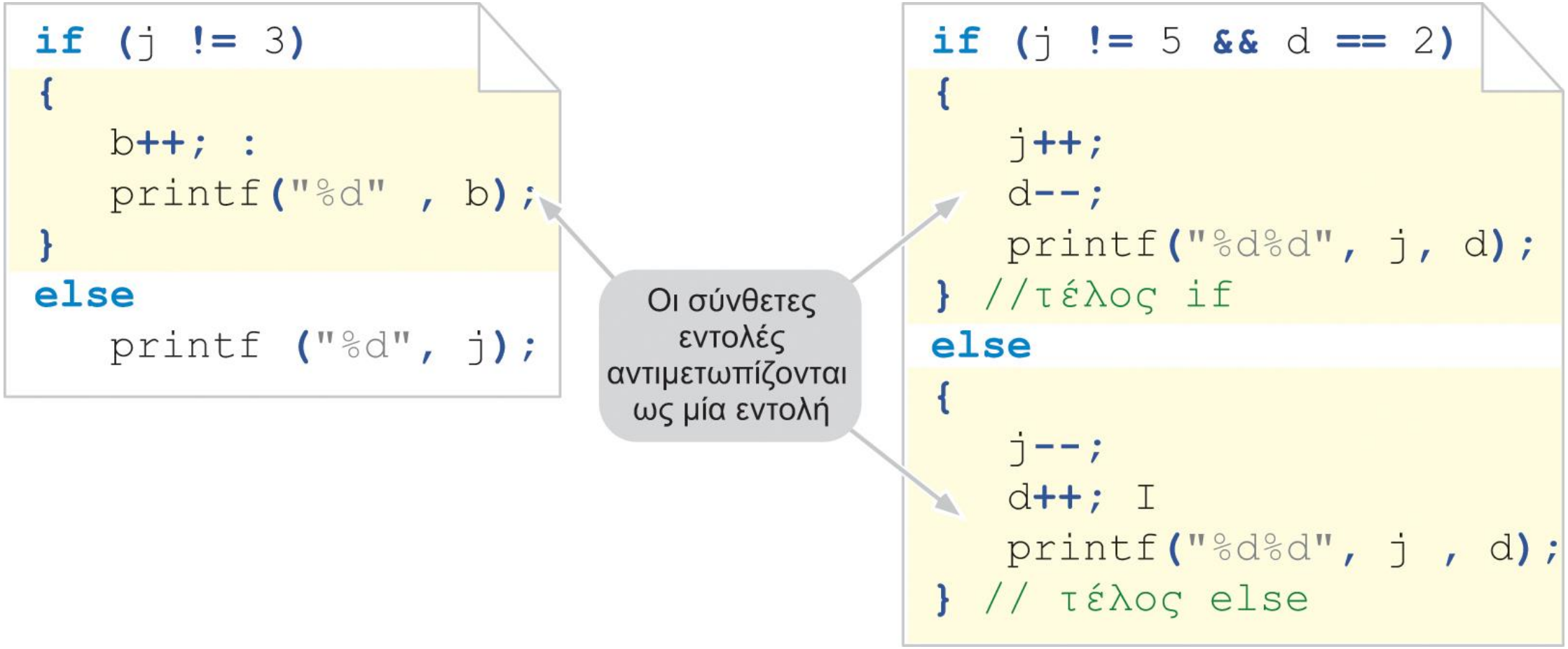


(α) Λογική ροή

```
if (έκφραση)
    εντολή1
else
    εντολή2
```

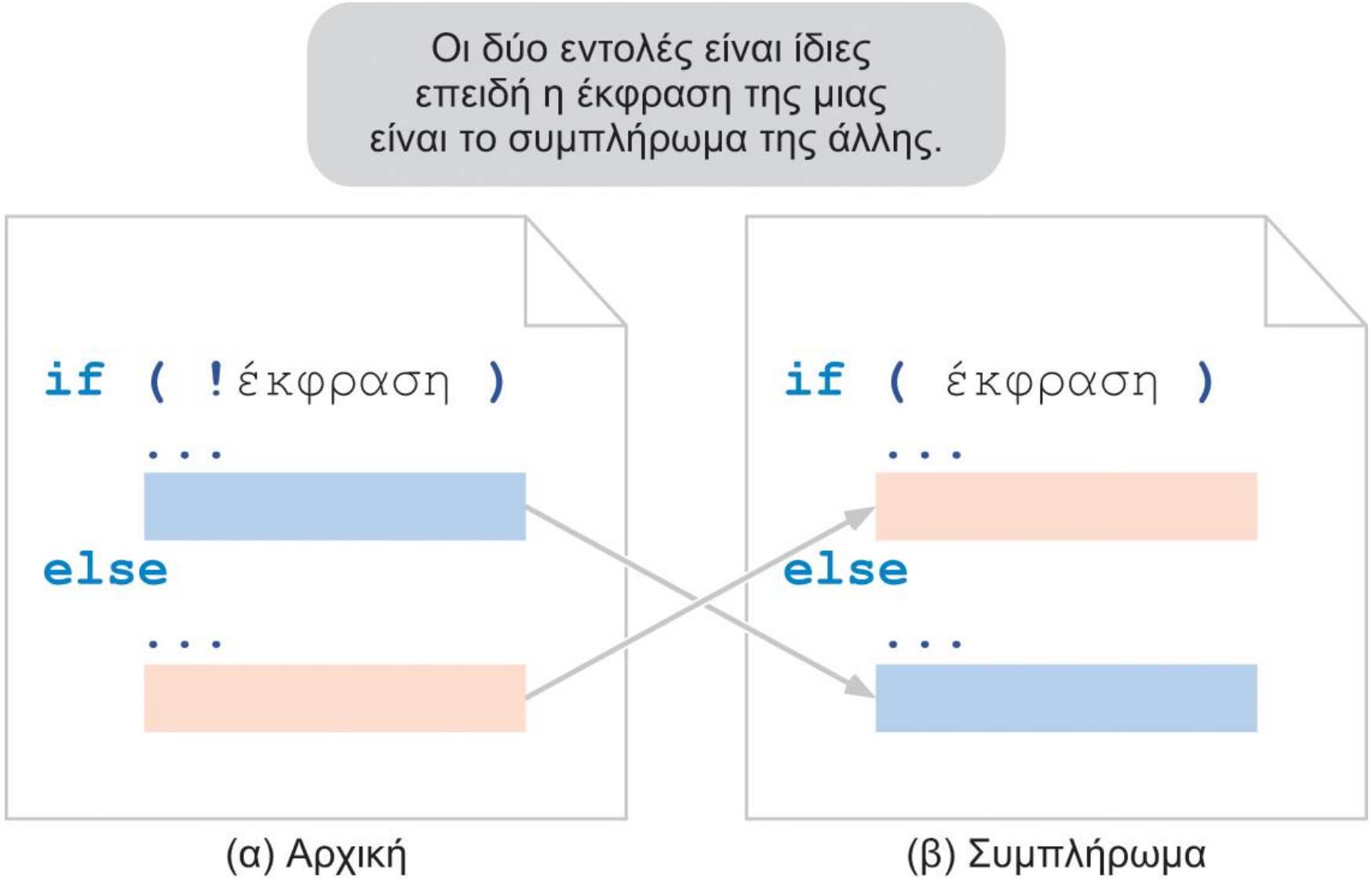
(β) Κώδικας

ΣΧΗΜΑ 5-7 Μία απλή εντολή if ... else.



ΣΧΗΜΑ 5-8 Χρήση σύνθετων εντολών σε μία if ... else.

ΣΧΗΜΑ 5-9
Συμπλήρωση έκφρασης και εναλλαγή εντολών στην εντολή if ... else.

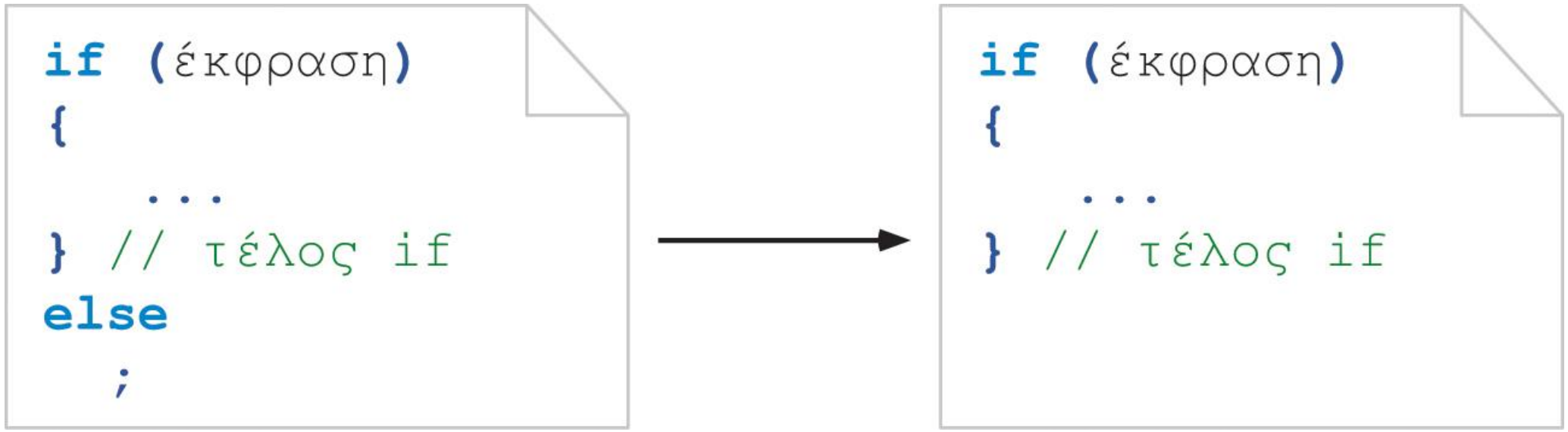


ΠΙΝΑΚΑΣ 5-4 Κανόνες σύνταξης για την εντολή if ... else

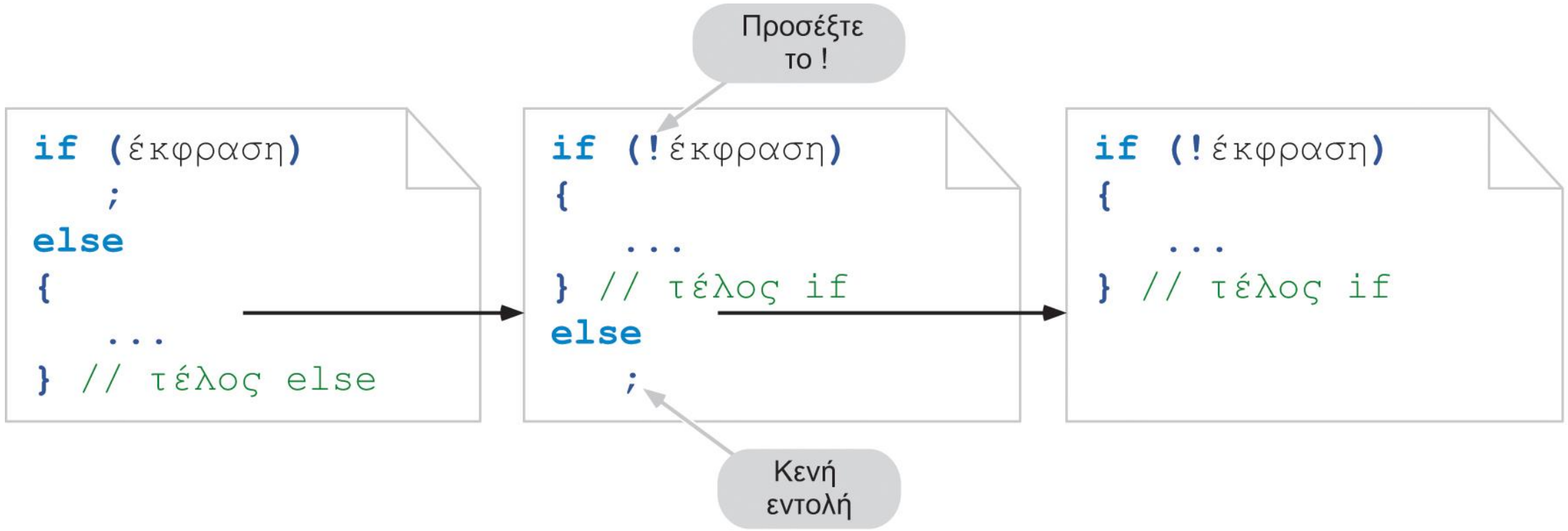
1.	Η έκφραση πρέπει να περικλείεται σε παρενθέσεις.
2.	Δεν απαιτείται χαρακτήρας τερματισμού (;) στις γραμμές εντολής if και else· τα τμήματα κώδικα που αντιπροσωπεύουν οι «εντολή1» και «εντολή2» μπορεί να απαιτούν χαρακτήρα τερματισμού, ανάλογα με τον τύπο τους.
3.	Η έκφραση της if μπορεί να έχει παρενέργεια.
4.	Η εντολή που πρόκειται να εκτελεστεί για κάθε ένα από τα δύο ενδεχόμενα της ελεγχόμενης έκφρασης (true και false) μπορεί να είναι οποιαδήποτε έγκυρη εντολή (ακόμα και μία επόμενη, ένθετη `if ... else`), ή μία κενή εντολή.
5.	Σε επίπεδο σύνταξης, οι «εντολή1» και «εντολή2» πρέπει να είναι μεμονωμένες εντολές. Ωστόσο, υπενθυμίζουμε ότι υπάρχει δυνατότητα συνδυασμού πολλαπλών εντολών σε μία μεμονωμένη, σύνθετη εντολή, με κατάλληλη χρήση των αγκίστρων ({...}).
6.	Μπορούμε να εναλλάξουμε τη θέση εκτέλεσης των «εντολή1» και «εντολή2», εάν αντί της αρχικής έκφρασης ελέγχου χρησιμοποιήσουμε το συμπλήρωμά της στην if.

Κενή (null) εντολή `else`

- Αν και υπάρχουν πάντα δύο πιθανές ενέργειες κατά τη λήψη μιας απόφασης, σε ορισμένες περιπτώσεις δεν μας ενδιαφέρουν και οι δύο
  - Όταν συμβαίνει αυτό, η «αδιάφορη» ενέργεια συνήθως παραλείπεται



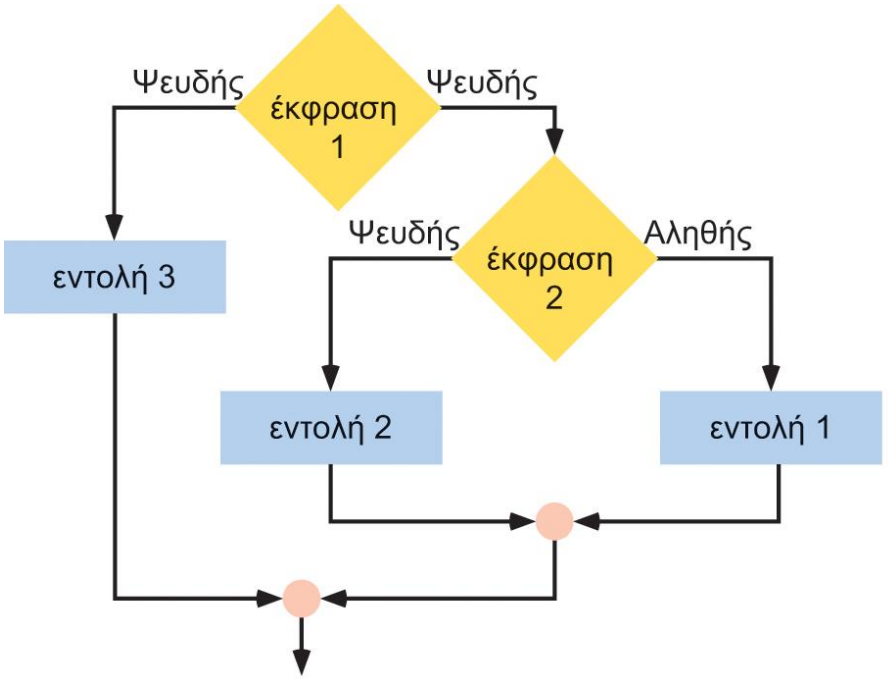
ΣΧΗΜΑ 5-10 Κενή εντολή `else`.



ΣΧΗΜΑ 5-11 Χρήση συμπληρώματος και κενής έκφρασης στην εντολή if ... else.

Ένθετες εντολές if

- Οι ενέργειες για τα δύο ενδεχόμενα μιας if ... else μπορούν να είναι οποιαδήποτε έγκυρη εντολή, συμπεριλαμβανομένης μιας επόμενης if ... else
- Όταν μία if ... else εμπεριέχεται σε μία άλλη if ... else, αναφέρεται ως ένθετη (nested) if



(α) Λογική ροή

```
if (έκφραση 1)
    if (έκφραση 2)
        εντολή 1
    else
        εντολή 2
else
    εντολή 3
```

(β) Κώδικας

ΣΧΗΜΑ 5-12 Ένθετες εντολές `if`.

Το πρόβλημα της «μετέωρης» else (1/3)

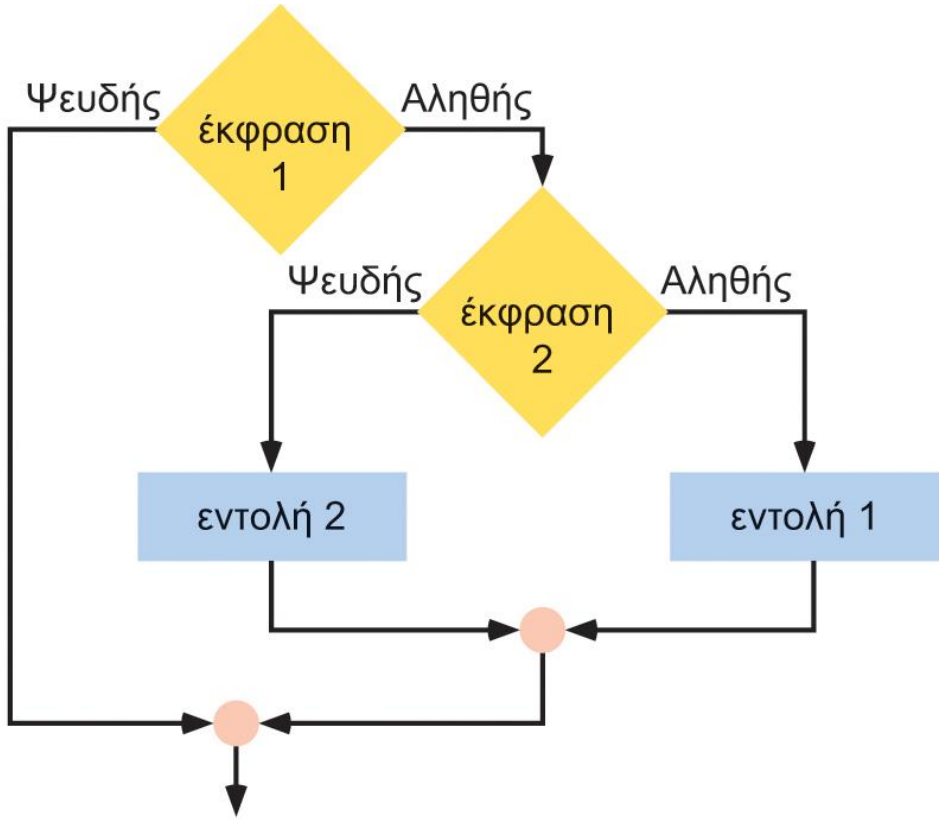
- Το πρόβλημα της «μετέωρης else» (dangling else) προκύπτει όταν δεν υπάρχει μία αντίστοιχη else για κάθε if, στην περίπτωση χρήσης ένθετων εντολών if ... else
- Η λύση που δίνει η C σε αυτό το πρόβλημα βασίζεται στον εξής απλό κανόνα:
  - μία else αντιστοιχίζεται πάντα στην πιο πρόσφατη-κοντινή (μη αντιστοιχισμένη) εντολή if, στο τρέχον τμήμα κώδικα

Το πρόβλημα της «μετέωρης» else (2/3)

```
if (έκφραση 1)
    if (έκφραση 2)
        εντολή 1
else
    εντολή 2
```

Ο μεταγλωττιστής συσχετίζει την αμέσως προηγούμενη `if` με την `else`.

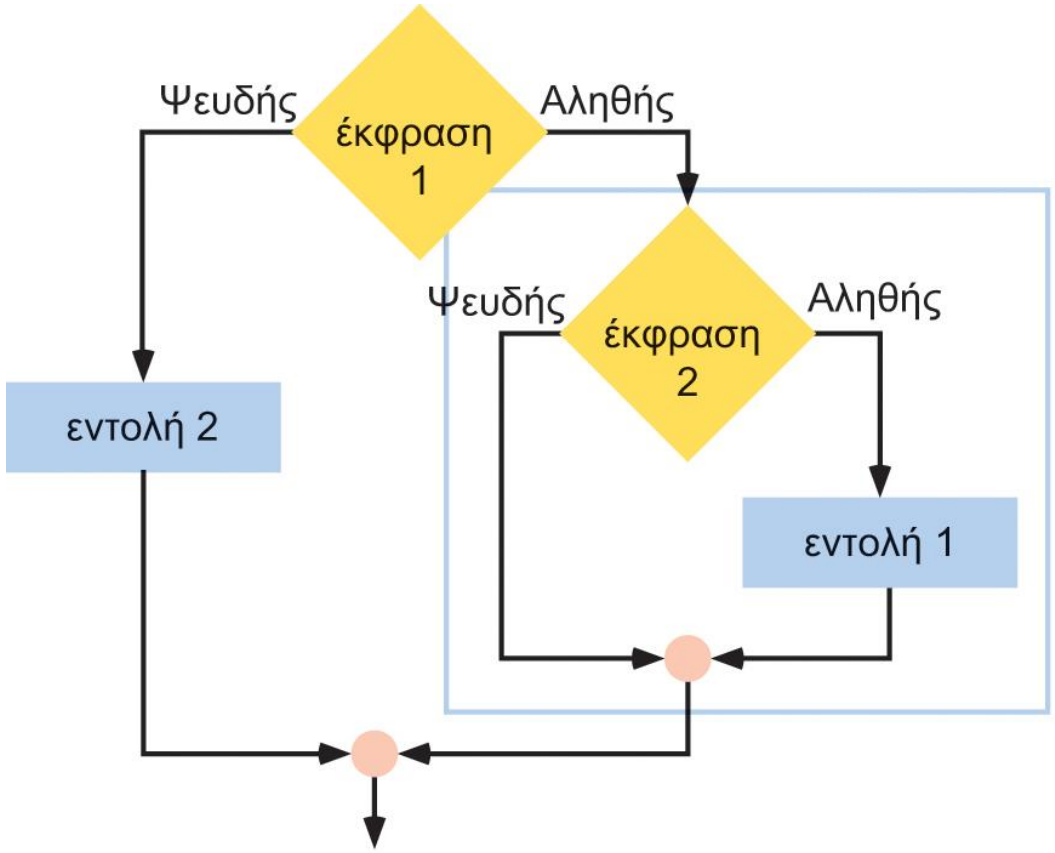
(α) Κώδικας



(β) Λογική ροή

ΣΧΗΜΑ 5-13 Το πρόβλημα της «μετέωρης» `else`.

Το πρόβλημα της «μετέωρης» else (3/3)



(α) Λογική ροή

Το μπλοκ ({...}) οριοθετεί την εντολή `if`

```
if (έκφραση 1)
{
    if (έκφραση 2)
        εντολή 1;
}
else
    εντολή 2;
```

(β) Κώδικας

ΣΧΗΜΑ 5-14 Λύση για το πρόβλημα της «μετέωρης» `else`.

Απλοποίηση εντολών `if`

- Οι εντολές if ... else μπορούν να γίνουν αρκετά πολύπλοκες
  - Συνήθως, ο σκοπός της απλοποίησης είναι να οδηγήσει σε πιο ευανάγνωστο και ευκολονόητο κώδικα

ΠΙΝΑΚΑΣ 5-5 Απλοποίηση της συνθήκης μιας εντολής `if`

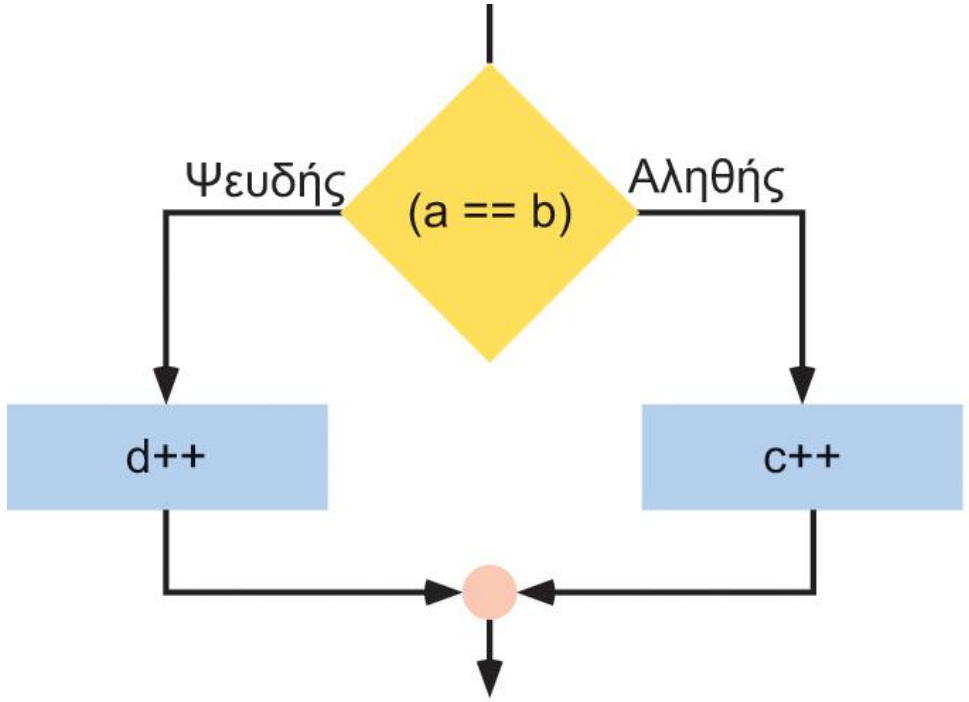
ΑΡΧΙΚΗ ΜΟΡΦΗ	ΑΠΛΟΠΟΙΗΜΕΝΗ ΜΟΡΦΗ
`if (a != 0)` `  εντολή1`	`if (a)` `  εντολή1`
`if (a == 0)` `  εντολή2`	`if (!a)` `  εντολή2`

Εκφράσεις υπό συνθήκη (1/2)

- Η C παρέχει μία βολική εναλλακτική λύση έναντι της «παραδοσιακής» χρήσης της εντολής if ... else για επιλογή δύο κατευθύνσεων, η οποία έχει τη μορφή μιας τριμερούς έκφρασης υπό συνθήκη, με προτεραιότητα 3
- Μία έκφραση υπό συνθήκη (conditional expression) περιλαμβάνει τρεις τελεστέους και έναν δυαδικό τελεστή
  - Κάθε τελεστέος είναι μία έκφραση
  - Το πρώτο σύμβολο του τελεστή (?) διαχωρίζει τις δύο πρώτες εκφράσεις και το δεύτερο σύμβολο του τελεστή (:) διαχωρίζει τις δύο τελευταίες εκφράσεις:

```
έκφραση ? έκφραση1 : έκφραση2
```

Εκφράσεις υπό συνθήκη (2/2)



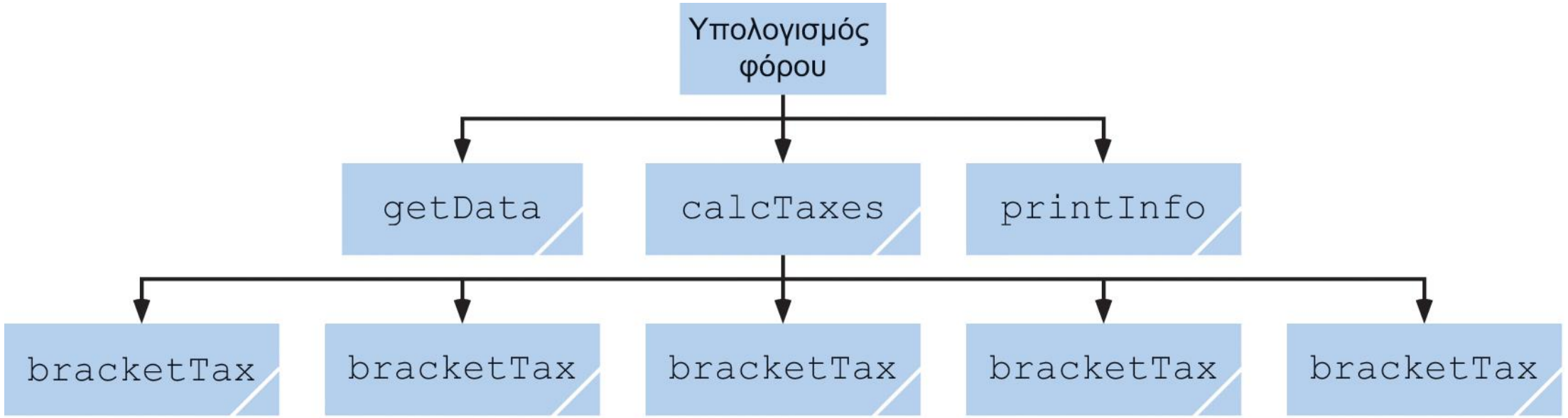
(α) Λογική ροή

```
a == b ? c++ : d--;
```

(β) Κώδικας

ΣΧΗΜΑ 5-15 Έκφραση υπό συνθήκη.

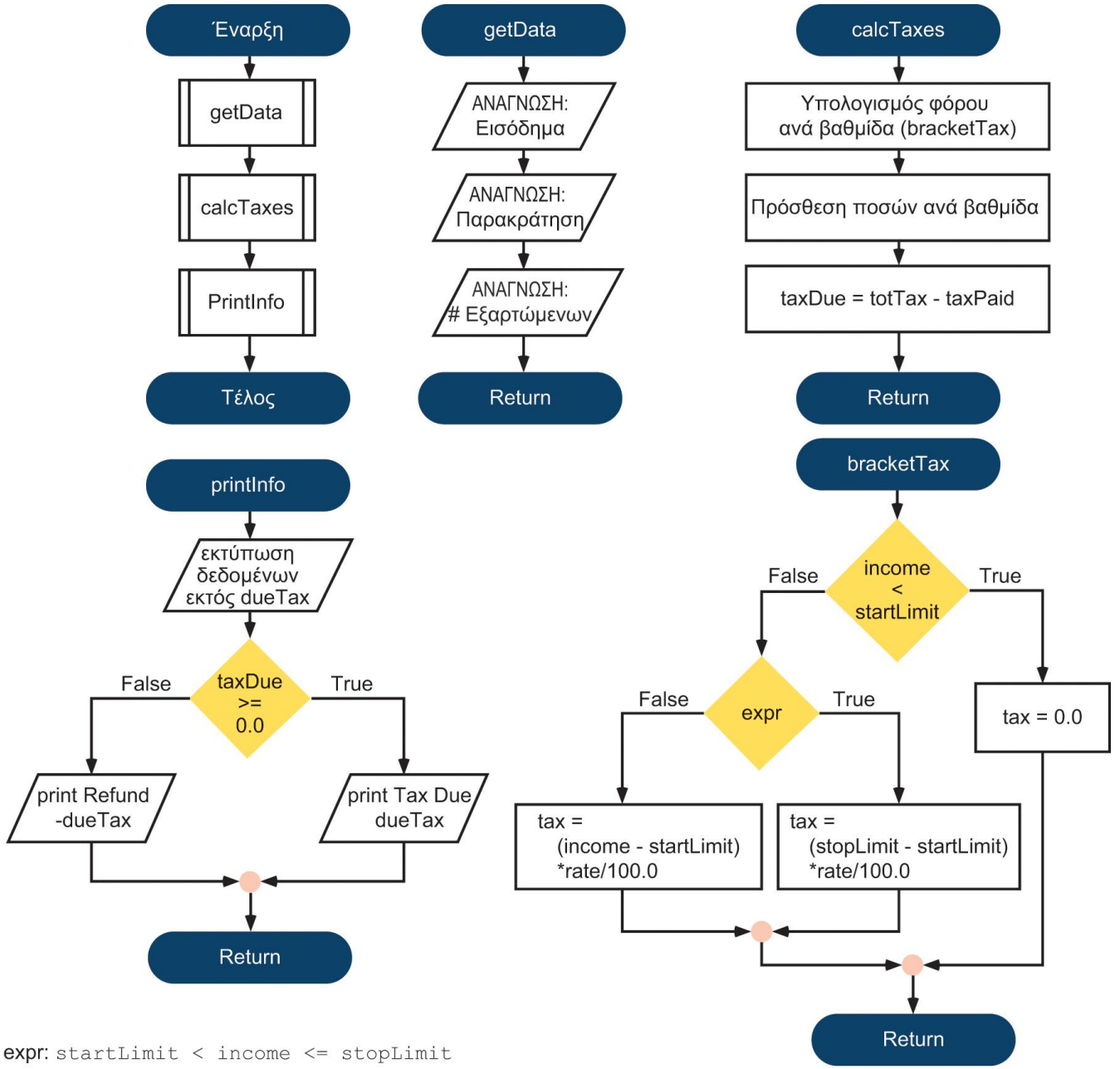
Παράδειγμα επιλογής δύο κατευθύνσεων



ΣΧΗΜΑ 5-16 Σχεδίαση του προγράμματος υπολογισμού φορολογίας.

Παράδειγμα επιλογής δύο κατευθύνσεων

ΣΧΗΜΑ 5-17 Σχεδίαση των συναρτήσεων του Προγράμματος 5-5.



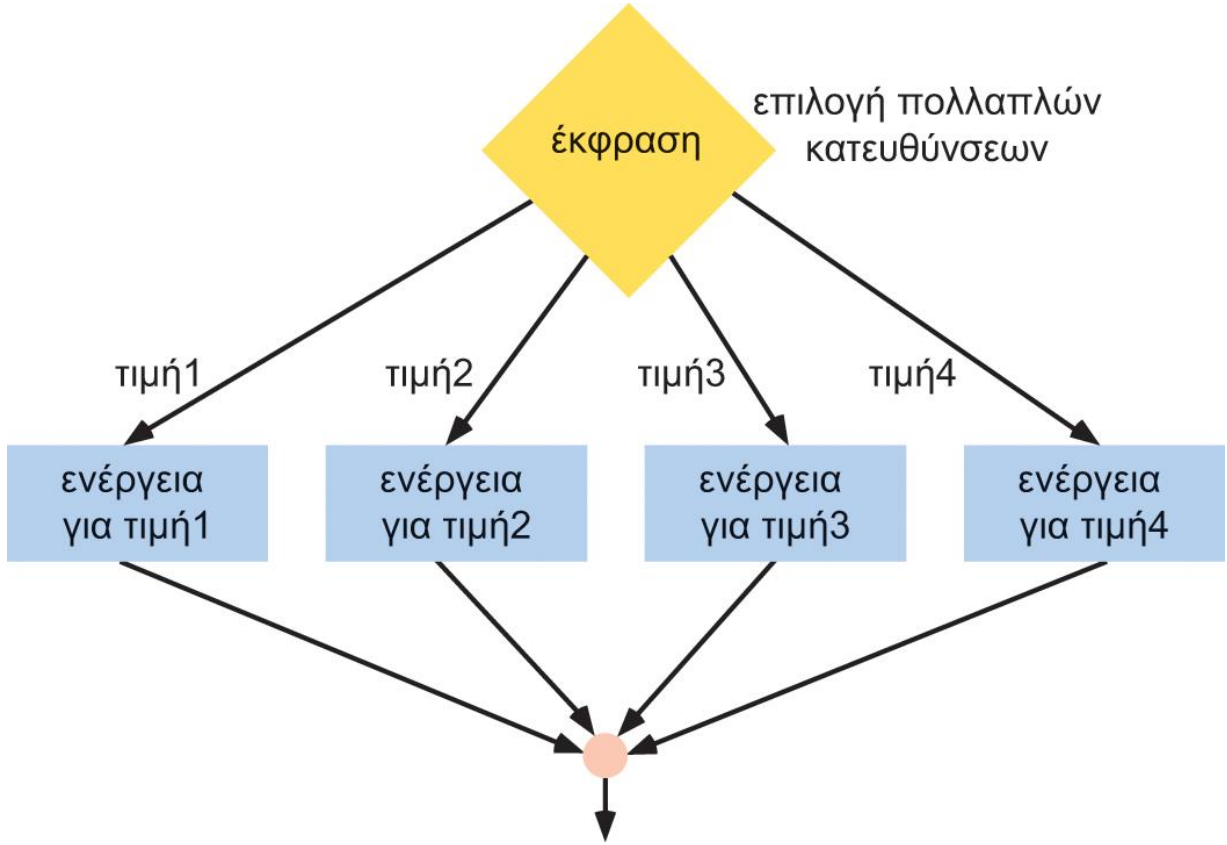
expr: `startLimit < income <= stopLimit`

5.3
Επιλογή πολλαπλών κατευθύνσεων

Επιλογή πολλαπλών κατευθύνσεων

- Οι περισσότερες γλώσσες προγραμματισμού παρέχουν δυνατότητα επιλογής πολλαπλών κατευθύνσεων
  - Δομές ή μηχανισμοί που επιτρέπουν λήψη αποφάσεων όταν υπάρχουν περισσότερες από δύο εναλλακτικές «περιπτώσεις»

ΣΧΗΜΑ 5-18 Λογική της διαδικασίας επιλογής πολλαπλών κατευθύνσεων.



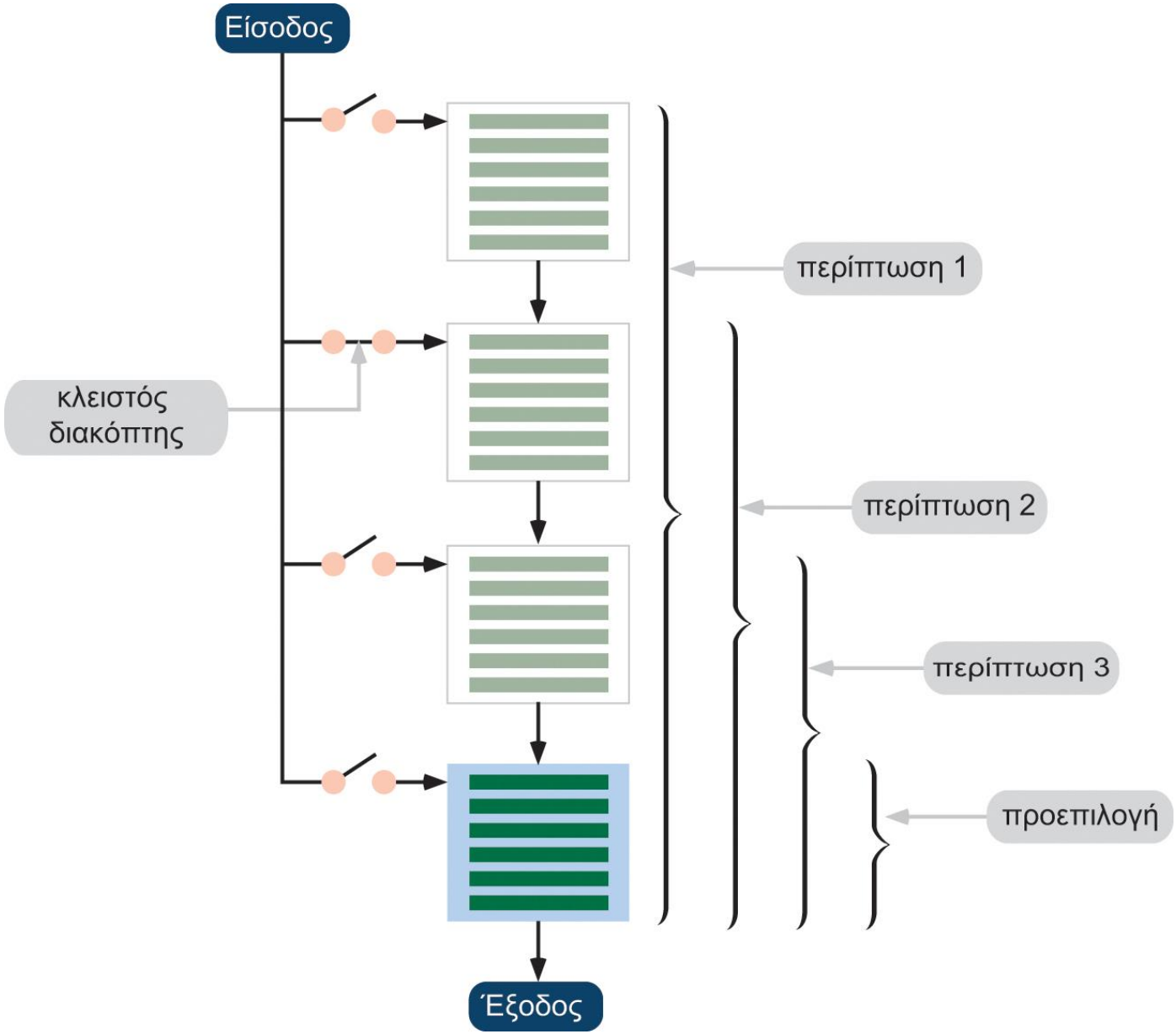
Η εντολή `switch` (1/3)

- Η εντολή switch είναι μία σύνθετη εντολή που χρησιμοποιείται για τη λήψη μιας απόφασης στην οποία εμπλέκονται πολλαπλές εναλλακτικές επιλογές ή «περιπτώσεις»

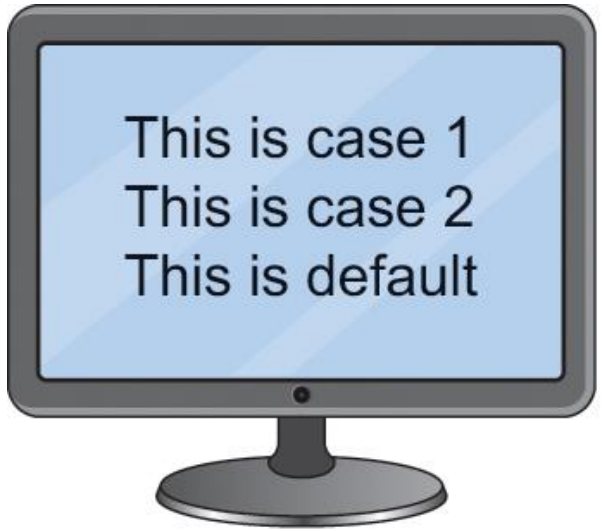
```
switch (έκφραση)
{
    case σταθερά -1 : εντολή
                      ...
                      εντολή
    case σταθερά -2 : εντολή
                      ...
                      εντολή
    case σταθερά -3 : εντολή
    default         : εντολή
                      ...
                      εντολή
} // τέλος switch
```

ΣΧΗΜΑ 5-19 Σύνταξη της εντολής `switch`.

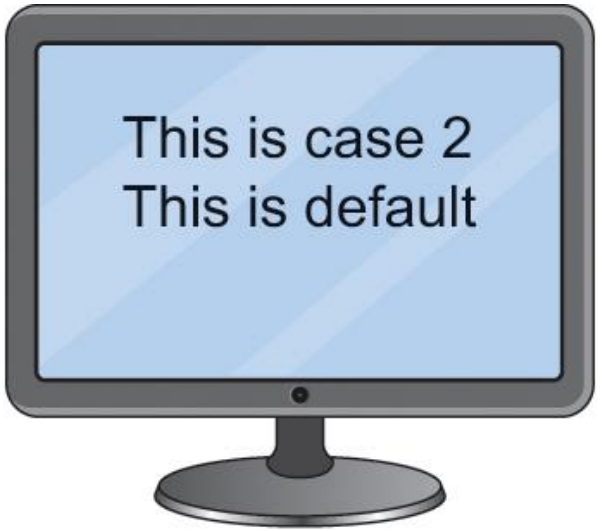
Η εντολή `switch` (2/3)



ΣΧΗΜΑ 5-20 Λογική της εντολής `switch`.



(α) Η `printFlag` έχει τιμή 1

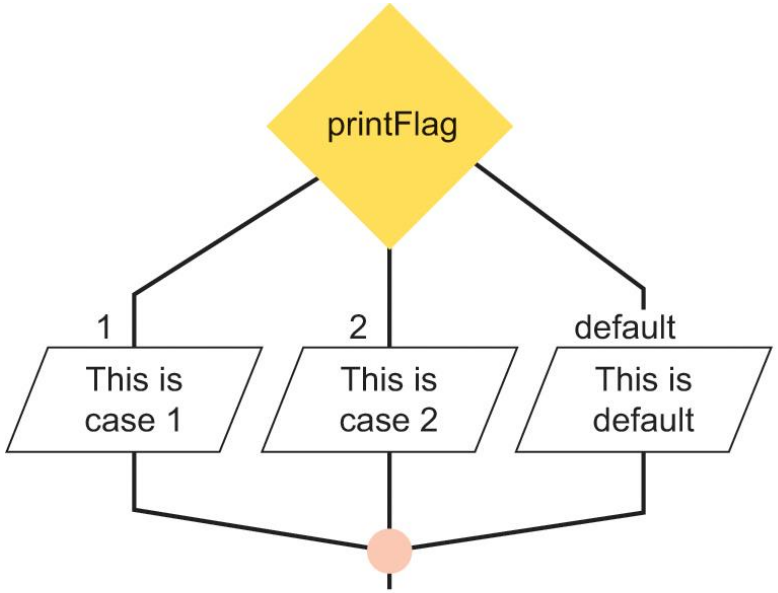


(β) Η `printFlag` έχει τιμή 2

(γ) Η `printFlag` δεν έχει τιμή 1 ή 2

ΣΧΗΜΑ 5-21 Πιθανά αποτελέσματα από την εκτέλεση της εντολής switch του Προγράμματος 5-6.

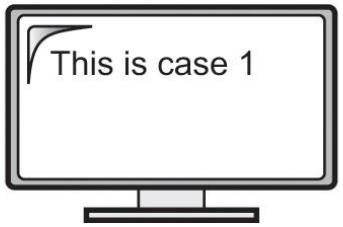
Η εντολή `switch` (3/3)

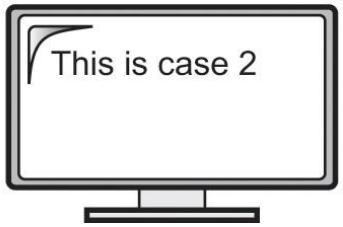


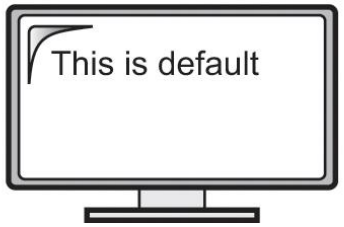
(α) Λογική ροή

```
switch (printFlag)
{
    case 1 : printf("This is case 1");
             break;
    case 2 : printf("This is case 2");
             break;
    default: printf("This is default");
             break;
} // τέλος switch
```

(β) Κώδικας



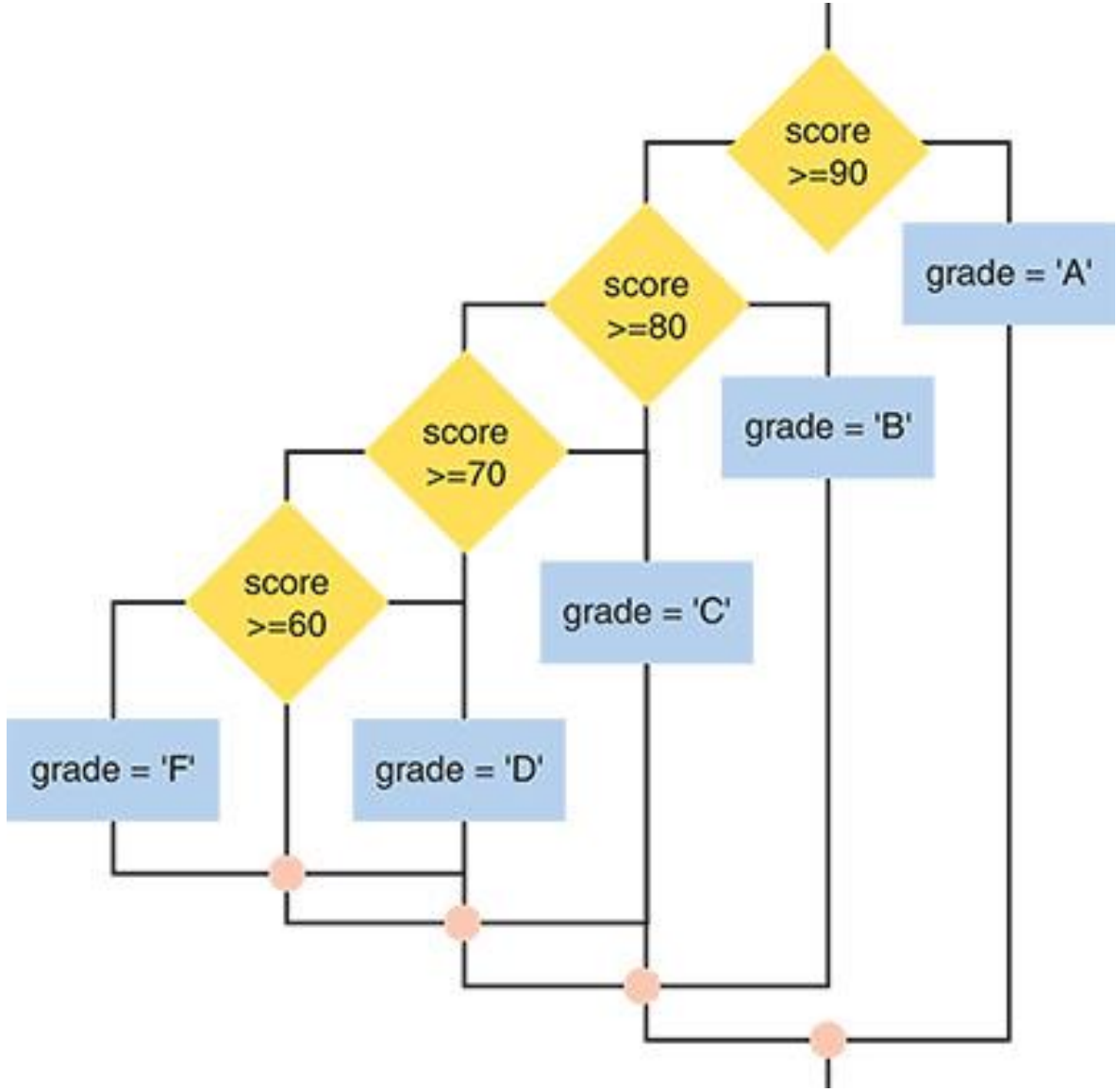




ΣΧΗΜΑ 5-22 Λογική ροή και σύνταξη μιας `switch` που περιλαμβάνει εντολές `break`.

Η εντολή `else … if`

- Ένας τρόπος γραφής κώδικα τον οποίο χρησιμοποιούμε σε περιπτώσεις όπου χρειαζόμαστε δυνατότητα επιλογής πολλαπλών κατευθύνσεων με βάση τιμές που δεν είναι ακέραιου τύπου



ΣΧΗΜΑ 5-23 Σχηματική αναπαράσταση της λογικής `else … if` για το Πρόγραμμα 5-9.

5.4
Άλλες συναρτήσεις της πρότυπης βιβλιοθήκης

Άλλες πρότυπες συναρτήσεις

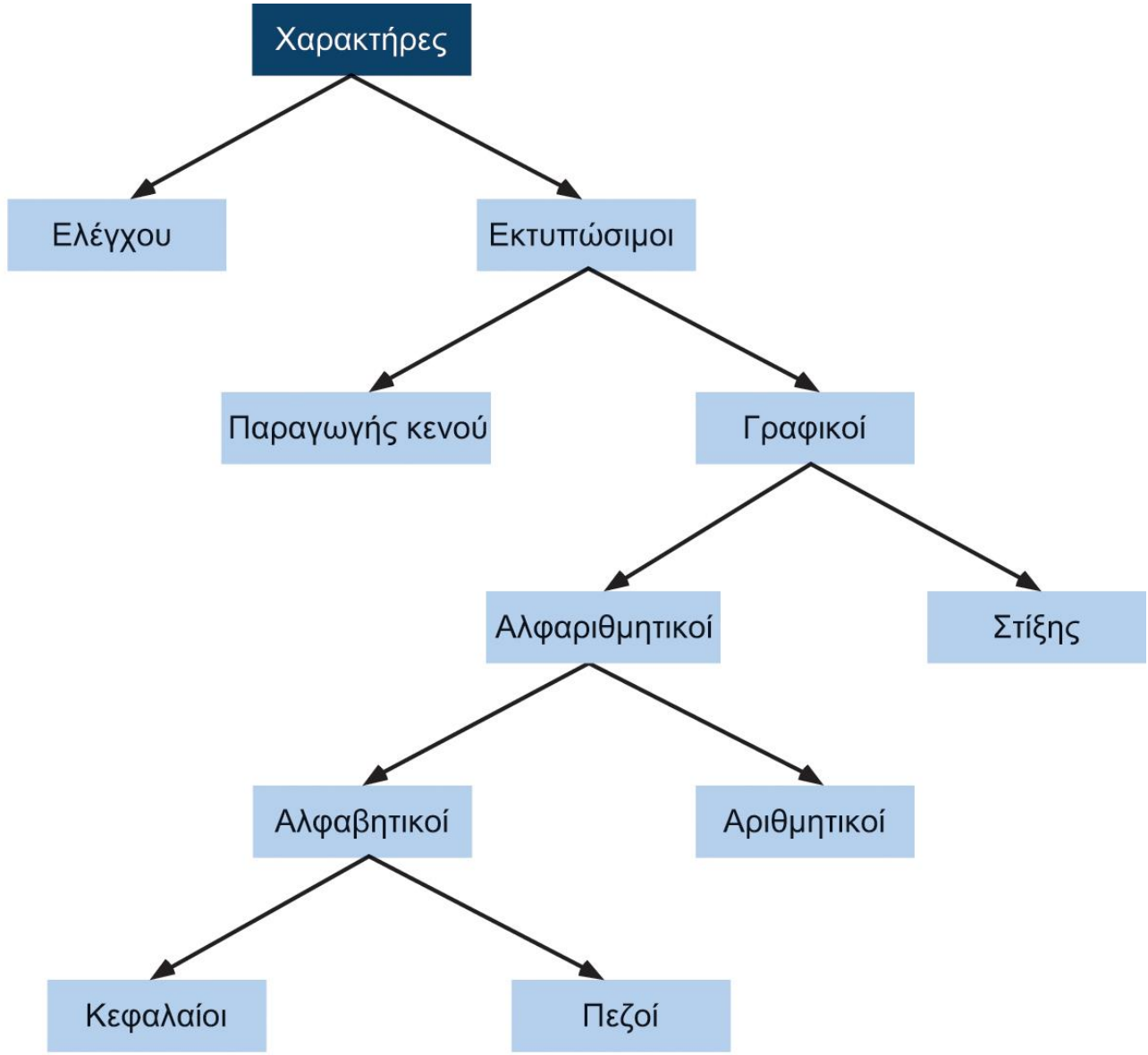
- Ένα από τα πλεονεκτήματα της γλώσσας C είναι το πλούσιο σύνολο πρότυπων συναρτήσεων που διευκολύνουν τον προγραμματισμό
- Από την έκδοση C18 και μετά, η C διαθέτει δύο αρχεία κεφαλίδας για τις συναρτήσεις χειρισμού χαρακτήρων
  - Το `ctype.h` περιλαμβάνει συναρτήσεις που υποστηρίζουν το σύνολο χαρακτήρων ASCII
  - Το `wctype.h` περιλαμβάνει συναρτήσεις που υποστηρίζουν σύνολα χαρακτήρων των 2-byte
    - wide characters: χαρακτήρες πλήρους πλάτους

Πρότυπες συναρτήσεις χειρισμού χαρακτήρων (1/2)

- Μπορούμε να διακρίνουμε τις συναρτήσεις χειρισμού χαρακτήρων σε δύο ευρείες κατηγορίες: συναρτήσεις ταξινόμησης και συναρτήσεις μετατροπής
- Όλες οι συναρτήσεις κατηγοριοποίησης χαρακτήρων επιστρέφουν μία λογική τιμή—εάν ο χαρακτήρας ανήκει στην κατηγορία για την οποία ελέγχει η συνάρτηση, η επιστρεφόμενη τιμή είναι true· διαφορετικά, είναι false
- Η C διαθέτει δύο συναρτήσεις οι οποίες χρησιμοποιούνται για τη μετατροπή της μορφής αναγραφής ενός χαρακτήρα—από κεφαλαίο σε πεζό, ή το αντίστροφο
  - Τα ονόματα αυτών των συναρτήσεων ξεκινούν με πρόθεμα to ή tow (για χαρακτήρες πλήρους πλάτους) και η επιστρεφόμενη τιμή τους είναι ένας ακέραιος που αντιστοιχεί στον μετατρεπόμενο χαρακτήρα)

Πρότυπες συναρτήσεις χειρισμού χαρακτήρων (2/2)

ΣΧΗΜΑ 5-24
Το σχήμα κατηγοριοποίησης χαρακτήρων που χρησιμοποιεί η C.



ΠΙΝΑΚΑΣ 5-8 Συναρτήσεις κατηγοριοποίησης χαρακτήρων

ΣΥΝΑΡΤΗΣΗ	ΠΕΡΙΓΡΑΦΗ
`iscntrl`	Ελέγχει για χαρακτήρες ελέγχου
`isprint`	Ελέγχει για εκτυπώσιμους χαρακτήρες (χαρακτήρες με απεικονίσιμη μορφή)
`isspace`	Ελέγχει για χαρακτήρες παραγωγής κενού: κενό διάστημα (32), οριζόντιος στηλοθέτης (9), κατακόρυφος στηλοθέτης (11), αλλαγή γραμμής (line feed, 10), αλλαγή σελίδας (form feed, 12) και επαναφορά σε αρχή γραμμής (carriage return, 13)
`isgraph`	Ελέγχει για «γραφικούς» χαρακτήρες, δηλαδή χαρακτήρες με απεικονίσιμη μορφή
`isalnum`	Ελέγχει για αλφαριθμητικούς χαρακτήρες: οποιοσδήποτε αλφαβητικός ή αριθμητικός χαρακτήρας
`ispunct`	Ελέγχει για χαρακτήρες που θεωρούνται σύμβολα στίξης: οποιοσδήποτε γραφικός χαρακτήρας που δεν είναι γράμμα ή ψηφίο
`isalpha`	Ελέγχει για αλφαβητικούς χαρακτήρες (κεφαλαίους ή πεζούς)
`isupper`	Ελέγχει εάν ένας αλφαβητικός χαρακτήρας είναι κεφαλαίος
`islower`	Ελέγχει εάν ένας αλφαβητικός χαρακτήρας είναι πεζός
`isdigit`	Ελέγχει εάν ένας χαρακτήρας είναι αρθμητικό ψηφίο (0 ... 9)
`isxdigit`	Ελέγχει εάν ένας χαρακτήρας είναι αρθμητικό ψηφίο στο δεκαεξαδικό σύστημα (0 ... 9, a ... f, A ... F)
`Isodigit`	Ελέγχει εάν ένας χαρακτήρας είναι αρθμητικό ψηφίο στο οκταδικό σύστημα (0 ... 7)

Ένα πρόγραμμα κατηγοριοποίησης χαρακτήρων

- Οι περισσότεροι από τους χαρακτήρες που υπάρχουν σε ένα τυπικό πληκτρολόγιο μπορούν να ενταχθούν σε περισσότερες από μία κατηγορίες
  - Για παράδειγμα, ένας αριθμός είναι ένας χαρακτήρας που εμπίπτει στις κατηγορίες «εκτυπώσιμος», «γραφικός», «αλφαριθμητικός» και «ψηφίο»

Χειρισμός σημαντικών σφαλμάτων

- GIGO: ένα από τα γνωστότερα ακρωνύμια στον κόσμο των υπολογιστών
  - Προέρχεται από την έκφραση Garbage In, Garbage Out που, σε ελεύθερη απόδοση, σημαίνει «τα λάθη στην είσοδο παράγουν λάθη στην έξοδο»
  - Όταν γράφουμε προγράμματα, είναι συνετό να αποφασίζουμε εξαρχής πώς θα χειριζόμαστε τα σφάλματα, τόσο αυτά που σχετίζονται με άκυρη είσοδο όσο και τα σφάλματα λογικής, προκειμένου να αποτρέψουμε το ενδεχόμενο εσφαλμένης απόκρισης από το πρόγραμμα

Χειρισμός σημαντικών σφαλμάτων

- Όταν δεν υπάρχει δυνατότητα ανάκαμψης από ένα (συνήθως σοβαρό) σφάλμα, η C παρέχει δύο συναρτήσεις, οι οποίες μας επιτρέπουν να τερματίσουμε άμεσα την εκτέλεση του κώδικα: exit και abort
  - Ενώ η εντολή return τερματίζει μία συνάρτηση «φυσιολογικά», η exit τερματίζει το πρόγραμμα άμεσα, ανεξάρτητα από τη θέση στην οποία βρίσκεται
  - Η συνάρτηση abort χρησιμοποιείται για μη ομαλό (ή «βίαιο» όπως λέγεται συνήθως) τερματισμό ενός προγράμματος

ΠΙΝΑΚΑΣ 5-10 Συμπληρωματικές εκφράσεις

ΑΡΧΙΚΗ ΕΝΤΟΛΗ	ΣΥΜΠΛΗΡΩΜΑΤΙΚΗ ΜΟΡΦΗ
`if (x <= 0)`	`if (x > 0)`
`if (x != 5)`	`if (x == 5)`
`if (!(x <= 0 || !flag))`	`if (x > 0 && flag)`

Σύνοψη

- Ένα στοιχείο δεδομένων χαρακτηρίζεται ως «λογικό» εάν φέρει την έννοια του αληθούς (true), ή του ψευδούς (false)
- Η C99 υλοποιεί τον τύπο bool για λογικά δεδομένα. Επιπλέον, υποστηρίζει τη χρήση ακεραίων για αναπαράσταση λογικών δεδομένων: εάν ένα στοιχείο δεδομένων είναι διάφορο του μηδενός, θεωρείται true· εάν είναι μηδέν, θεωρείται false
- Η C διαθέτει τρεις τελεστές για την εκτέλεση λογικών πράξεων: *not* (άρνηση), *and* (σύζευξη) και *or* (διάζευξη)
- Η C διαθέτει έξι τελεστές για πράξεις σύγκρισης: <, <=, >, >=, == και !=
- Η διαδικασία επιλογής στη C υλοποιείται με χρήση δύο εντολών: if ... else (επιλογή δύο κατευθύνσεων) και switch (επιλογή πολλαπλών κατευθύνσεων)

Σύνοψη

- Η δομή if ... else χρησιμοποιείται για την επιλογή μεταξύ δύο εναλλακτικών «περιπτώσεων»
- Μπορείτε να εναλλάξετε τον κώδικα που εκτελείται για τις δύο εναλλακτικές περιπτώσεις μιας εντολής if ... else, χρησιμοποιώντας το συμπλήρωμα της έκφρασης ελέγχου στην if
- Εάν σε μία εντολή if ... else δεν απαιτείται χειρισμός της περίπτωσης όπου η έκφραση ελέγχου αποτιμάται σε false, μπορεί να παραληφθεί το τμήμα else
- Εάν εντοπιστεί μία «μετέωρη» else, θα αντιστοιχιστεί στην τελευταία (πιο πρόσφατη) εντολή if που δεν αντιστοιχίζεται ήδη σε μία else
- Στη C, η επιλογή πολλαπλών κατευθύνσεων μπορεί να υλοποιηθεί με χρήση είτε της εντολής switch, είτε μιας δομής else ... if

Σύνοψη

- Η εντολή switch χρησιμοποιείται για την επιλογή ή λήψη απόφασης μεταξύ πολλών εναλλακτικών «περιπτώσεων», όταν οι συνθήκες ελέγχου που καθοδηγούν τη διαδικασία επιλογής μπορούν να εκφραστούν ως ακέραιες τιμές
- Η δομή else ... if χρησιμοποιείται για την επιλογή/λήψη απόφασης μεταξύ πολλών εναλλακτικών «περιπτώσεων», όταν το ελεγχόμενο στοιχείο που καθοδηγεί τη διαδικασία επιλογής δεν είναι ακέραιου τύπου και, επομένως, δεν μπορεί να χρησιμοποιηθεί εντολή switch
- Για τον καθορισμό των εναλλακτικών περιπτώσεων σε μία εντολή switch χρησιμοποιείται η δεσμευμένη λέξη case

Σύνοψη

- Για τον καθορισμό μιας «προεπιλεγμένης περίπτωσης» σε μία εντολή switch χρησιμοποιείται η δεσμευμένη λέξη default. Εκτελείται όταν καμία από τις εναλλακτικές case δεν ταιριάζουν με την ελεγχόμενη τιμή και αναγράφεται τελευταία, μετά από όλες τις case

- Η αναγραφή σε κατάλληλο επίπεδο εσοχής όλων των εντολών που τελούν υπό τον έλεγχο μιας συνθήκης αποτελεί καλή πρακτική προγραμματισμού και διευκολύνει την ανάγνωση/κατανόηση του κώδικα

- Σε ένα διάγραμμα δομής, η διαδικασία επιλογής υποδεικνύεται μόνο όταν περιλαμβάνει κλήση σε άλλη συνάρτηση

Σύνοψη

● Η αποτύπωση της επιλογής στο διάγραμμα δομής υποδεικνύει τις διαδρομές που μπορεί να ακολουθήσει η λογική ροή.

● Δεν είναι πάντοτε δυνατόν να διακρίνουμε, με απλή εξέταση του διαγράμματος δομής, ποια διαδικασία επιλογής θα χρησιμοποιηθεί (διπλής ή πολλαπλής κατεύθυνσης)

- α. Η επιλογή υπό τη συνθήκη μιας εντολής if υποδεικνύεται με έναν ρόμβο κάτω από την καλούσα συνάρτηση
- β. Η επιλογή αμοιβαία αποκλειόμενων εναλλακτικών περιπτώσεων που υλοποιείται με χρήση μιας δομής if ... else ή switch υποδεικνύεται με ένα σύμβολο συν(+) ανάμεσα στις αμοιβαία αποκλειστικές διεργασίες/συναρτήσεις

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.