Δομημένος Προγραμματισμός

Παρουσίαση του μαθήματος

Ελευθερία Κατσίρη
Επίκουρη Καθηγήτρια

Μάθημα στο E-Class (https://eclass.duth.gr/)

Οργάνωση μαθήματος

- **Διαλέξεις:**
  - Παρασκευή 12μμ-2μμ
  - Κυρίως θεωρία με παραδείγματα/ασκήσεις
- **Εργαστήριο-Ασκήσεις: Υποχρεωτικά!**
  - Παρασκευή 4μμ-7μμ
  - “hands-on” ασκήσεις
  - Λάπτοπ, Smart Phone
  - C Compiler π.χ Visual Studio Code
- **Επικοινωνία**
  - Γραφτείτε στο e-class! **https://eclass.duth.gr**
  - Ώρες γραφείου: Τετάρτη 4μμ-6μμ

Δικαίωμα για μια απουσία!

Συγγράμματα

- **ΟΔΗΓΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΤΗ ΓΛΩΣΣΑ C,**

Α. Καράκος, Εκδόσεις Καράκος, 2017, Ξάνθη, ISBN: 978-618-83086-0-2. Κωδικός βιβλίου στον Εύδοξο: 68371534

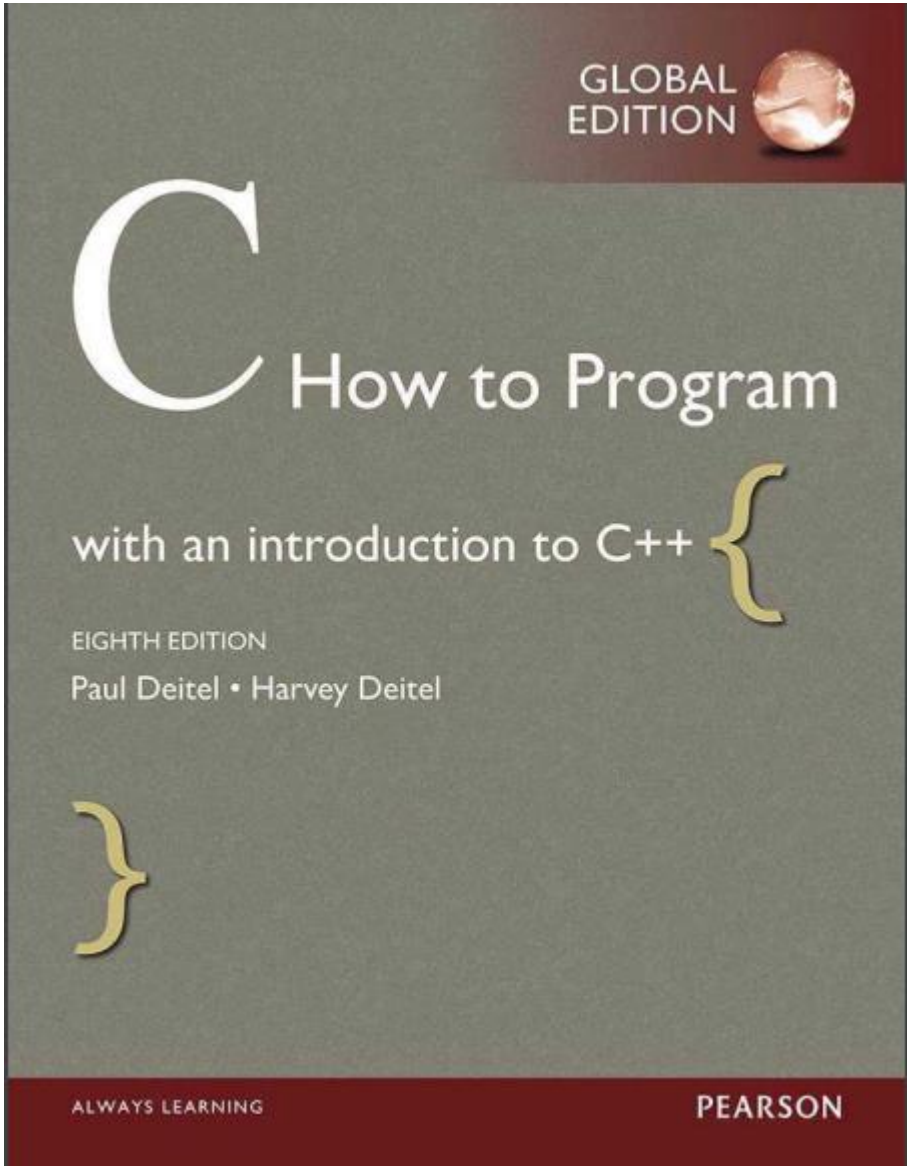
- **Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C,**

Kernighan, Ritchie, 2η έκδοση, Κωδικός Βιβλίου στον Εύδοξο: 22701978

✓ Θεωρείται «Το» βιβλίο για τη C. Ο συγγραφέας είναι από τους συγγραφείς της γλώσσας.

✓ Αλλά: Δεν είναι για αρχαρίους, προυποθέτει κάποια εξοικοίωση με τις βασικές αρχές προγραμματισμού

Συγγράμματα 2



Αξιολόγηση μαθήματος

- **Γραπτή εξέταση 50% βαθμού**
  - 1-2 προγράμματα,
- **Εξέταση εργαστηρίου 50% βαθμού**
  - Αλληλοδιόρθωση προγραμμάτων.
- **Δικαίωμα συμμετοχής στη γραπτή εξέταση:**

N-1 παρουσίες στα εργαστήρια, όπου N=#εργαστηρίων
AND
Βαθμός εξέτασης εργαστηρίου > 5

Δομημένος Προγραμματισμός

- **Στόχος του μαθήματος**

Η διδασκαλία και η εκμάθηση της γλώσσας προγραμματισμού C.

- **Γιατί η γλώσσα C;**

Η γλώσσα C αποτελεί τη βασικότερη γλώσσα για την ανάπτυξη εφαρμογών σε όλο το φάσμα της Πληροφορικής, γι αυτό το λόγο μπορεί να βοηθήσει στην εύκολη κατανόηση και εκμάθηση και άλλων γλωσσών προγραμματισμού που προβλέπονται στο πρόγραμμα σπουδών.

Μαθησιακοί Στόχοι

- **Πρώτο μάθημα προγραμματισμού με χρήση της γλώσσας C**
  - Δεν απαιτείται προηγούμενη γνώση!
- **Εως το τέλος του μαθήματος θα μπορείτε:**
  - Να κατανοήσετε θεμελιώδεις έννοιες των διαδικασιακών γλωσσών προγραμματισμού (procedural languages)
  - Να μπορείτε να σχεδιάσετε αλγορίθμους για να επιλύετε απλά προβλήματα
  - Να χρησιμοποιείτε τη γλώσσα C

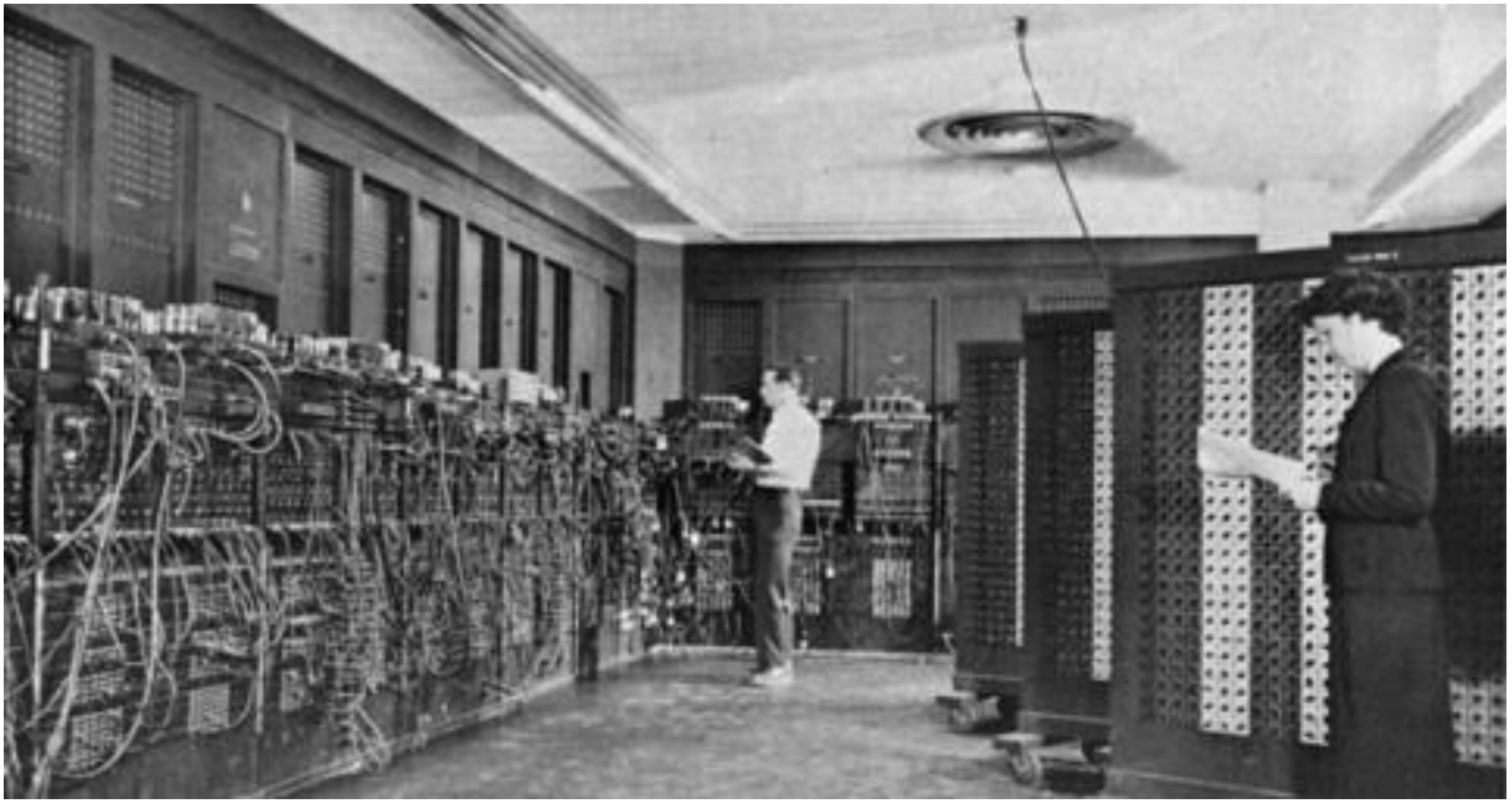
Ύλη μαθήματος

- Θεμελιώδεις Έννοιες
- Μεταγλωτίστε και τρέξτε το πρώτο πρόγραμμα σε C
- Μεταβλητές, τύποι δεδομένων και αριθμητικές εκφράσεις
- Βρόχοι επανάληψης
- Εντολές λήψης απόφασης
- Πίνακες
- Συναρτήσεις
- Δομές
- Αλφαριθμητικά
- Δείκτες

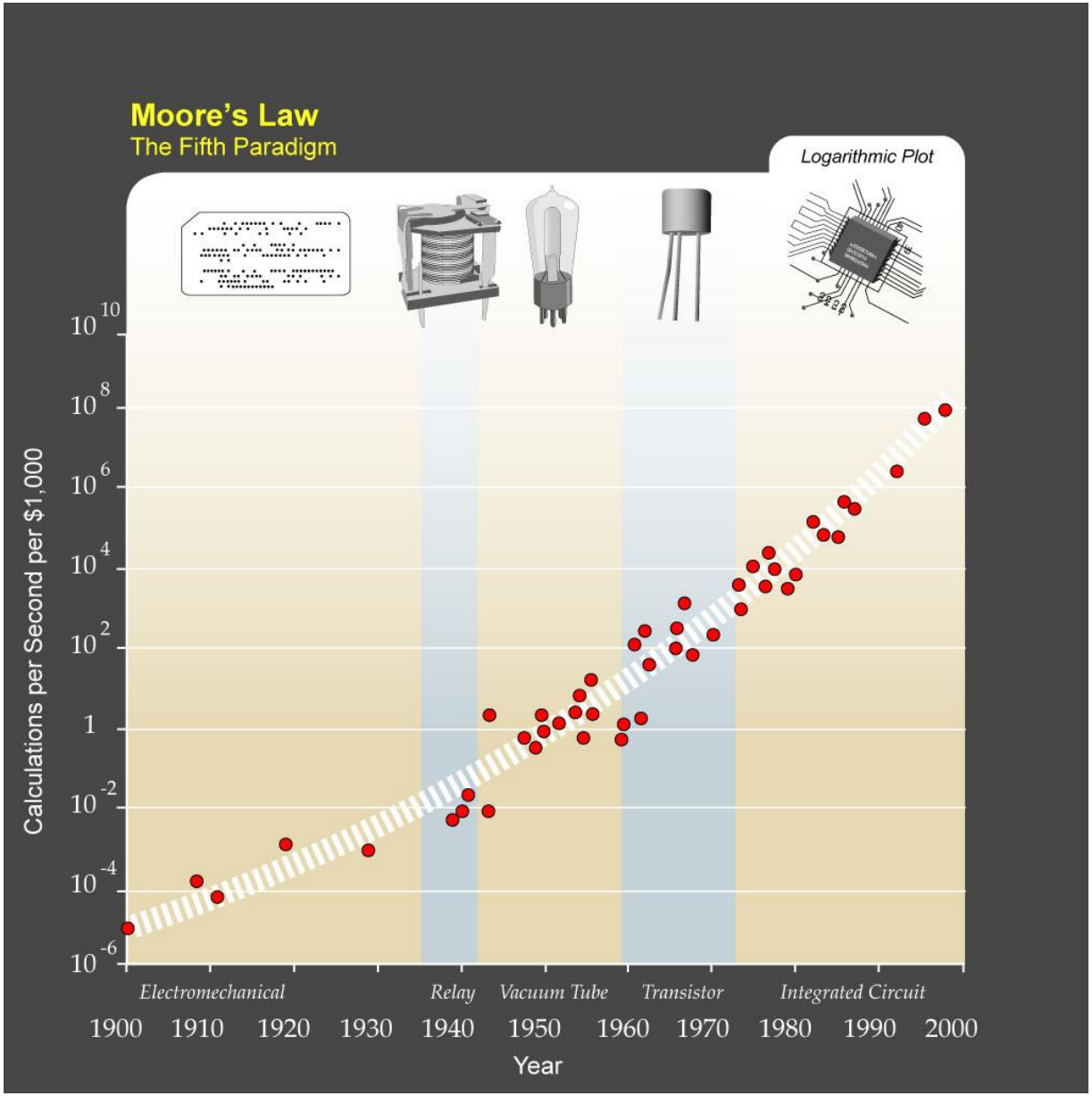
Θεμελιώδεις Έννοιες

- Μοντέλο για υπολογιστικές μηχανές
- Προγραμματισμός
- Γλώσσες Προγραμματισμού
- Μεταγλώττιση
- Λειτουργικά Συστήματα

Παρελθόν

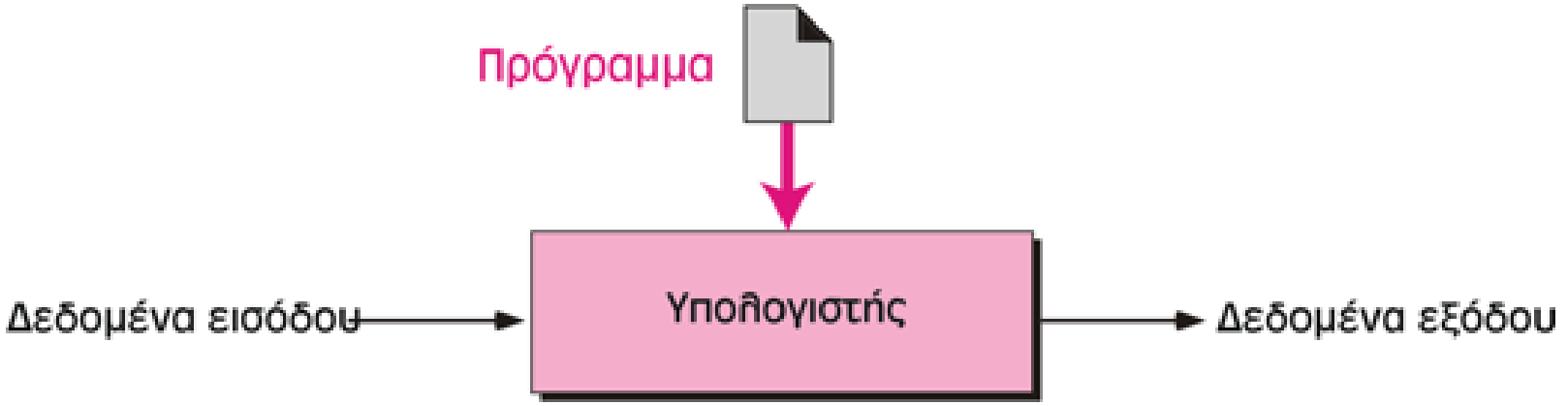


Ο Νόμος του Moore



Παρόν

Προγραμματιζόμενος Επεξεργαστής Δεδομένων



- **Πρόγραμμα** (Program) είναι ένα σύνολο οδηγιών / εντολών οι οποίες λένε στον υπολογιστή τι να κάνει με τα δεδομένα.
- Τα προγράμματα αποτελούνται από σύνολα εντολών οι οποίες είναι γραμμένες σε μια γλώσσα προγραμματισμού.
- Σ' αυτό το μοντέλο, τα δεδομένα εξόδου (output data) εξαρτώνται από το συνδυασμό δύο παραγόντων:
  - των δεδομένων εισόδου (input data)
  - του προγράμματος.

Πώς δουλεύει

- Πώς εκτελεί ένας υπολογιστής ένα πρόγραμμα; (π.χ. Ένα παιχνίδι, έναν επεξεργαστή κειμένου, κλπ)
- Οι εντολές του προγράμματος αντιγράφονται από τη μόνιμη δευτερεύουσα μνήμη στην κύρια μνήμη.
- Αφού φορτωθούν οι εντολές, η ΚΜΕ αρχίζει την εκτέλεση του προγράμματος.
- Κάθε εντολή **προσκομίζετα***ι* αρχικά από τη μνήμη, **αποκωδικοποιείται** για να καταλάβει η ΚΜΕ τι αντιπροσωπεύει, και **εκτελείται** η κατάλληλη ενέργεια (the *fetch, decode, execute cycle)*
- Και ούτω καθέξής η επόμενη εντολή.

Γλώσσα μηχανής

- Οι εντολές αντιπροσωπεύονται με δυαδική αρίθμηση (ακολουθίες από 0 , 1)
- Γιατί δυαδική; Γιατί το υλικό του υπολογιστή βασίζεται σε ηλεκτρικά/ηλεκτρονικά κυκλώματα τα οποία μπορούν να αλλάζουν μεταξύ δύο καταστάσεων / bits
- Το δυαδικό πρόγραμμα που θα τρέξει στην ΚΜΕ ενός επεξεργαστή θα μπορούσε να είναι το εξής: 1010 1111 0011 0111 0111 0110
- Οι πρώτοι υπολογιστές προγραμματίζονταν έτσι !
- Η δουλειά των πρώτων προγραμματιστών ήταν να γράφουν κώδικα κατευθείαν σε γλώσσα μηχανής χρησιμοποιώντας διακόπτες.

Ιστορικά στοιχεία για τη C

- Αναπτύχθηκε από τον Dennis Ritchie – 1972
- Προέρχεται από τη γλώσσα BCPL (Martin Richards, 1967) και τη γλώσσα B (Ken Thompson, 1970).
- Χρησιμοποιήθηκε για να γραφτεί το UNIX.
- Η γρήγορη ανάπτυξή της δημιούργησε πολλές παραλλαγές.
- Το 1989 καθιερώνεται η ANSI C
- Το 1999 η τελευταία επίσημη έκδοση C99
- Το 2011 προτείνεται η C11
- Το 2018 προτείνεται η C17
- Το 2024 προτείνεται η C23 (preview)

Βασικές γνώσεις για τη C

Οι *γλώσσες χαμηλού επιπέδου* αναγκάζουν τους προγραμματιστές να ορίζουν κατευθείαν όλες τις συναρτήσεις διότι τίποτε δεν είναι έτοιμο.

Οι γλώσσες *υψηλού επιπέδου* είναι έτσι σχεδιασμένες ώστε να παρέχουν στους προγραμματιστές όλες τις εντολές που επιθυμούν για να χρησιμοποιήσουν στην ανάπτυξη του κώδικα.

Μια γλώσσα *μέσου επιπέδου* δίνει στους προγραμματιστές ένα ελάχιστο σύνολο εντολών που μπορεί να χρησιμοποιηθεί για να δώσει δομές γλώσσας υψηλού επιπέδου.

Βασικές γνώσεις για τη C

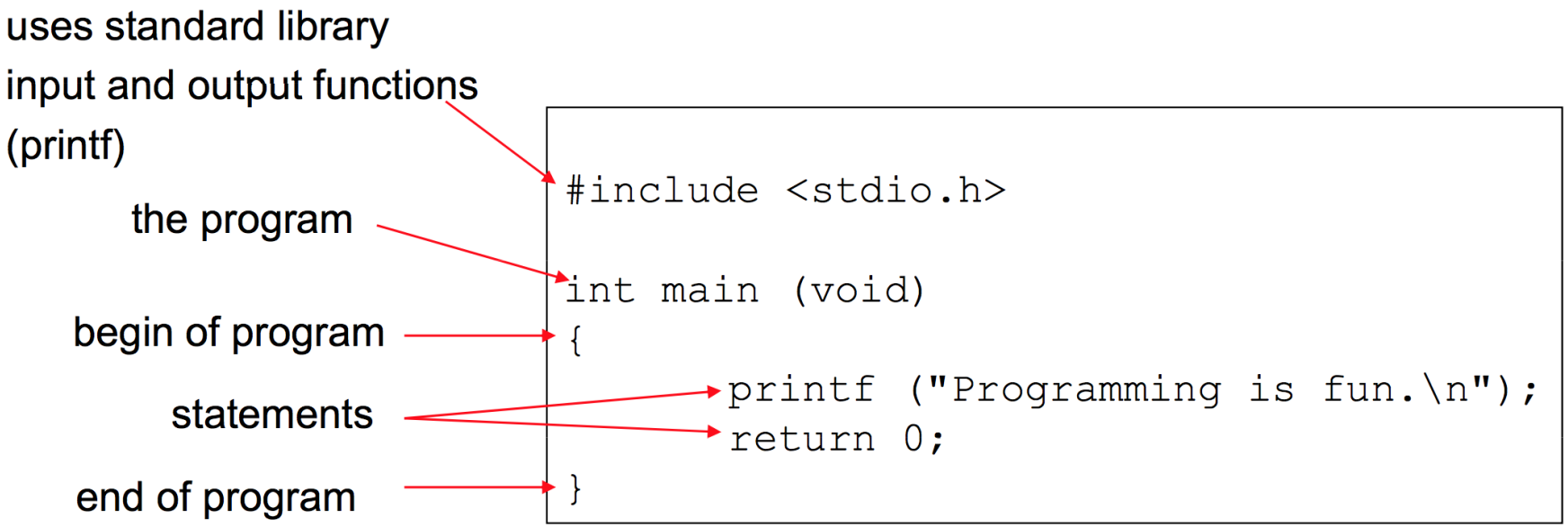
Κύριο χαρακτηριστικό γνώρισμα των γλωσσών μέσου επιπέδου είναι η εύκολη πρόσβαση στη μνήμη και στη ΚΜΕ (CPU), ενώ παράλληλα διατηρούν τον έλεγχο και τη δομή των δεδομένων όπως οι γλώσσες υψηλού επιπέδου.

Η πρόσβαση στη μνήμη επιτρέπει την ολίσθηση και την επεξεργασία των bits με τη χρησιμοποίηση ενός αρκετά πλούσιου συνόλου τελεστών.

- *Η γλώσσα C διαθέτει όλες αυτές τις ιδιότητες*

γι αυτό και έγινε γρήγορα δημοφιλής ανάμεσα στους προγραμματιστές και κυρίως στους προγραμματιστές λειτουργικών συστημάτων.

The first C program



main: a special name that indicates where the program must begin execution. It is a special *function*.

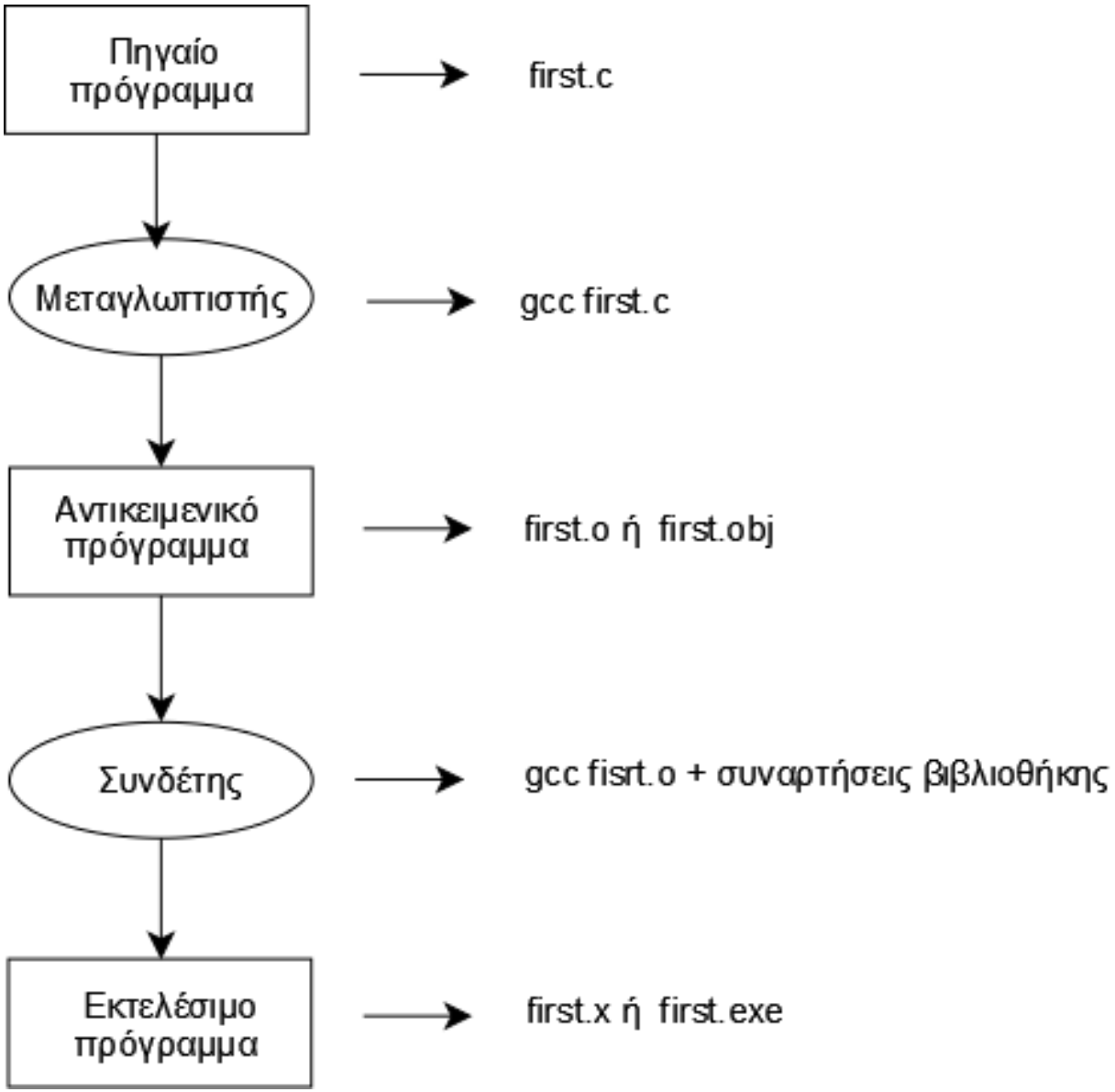
first statement: calls a routine named printf, with argument the string of characters “Programming is fun \n”

last statement: finishes execution of main and return to the system a status value of 0 (conventional value for OK)

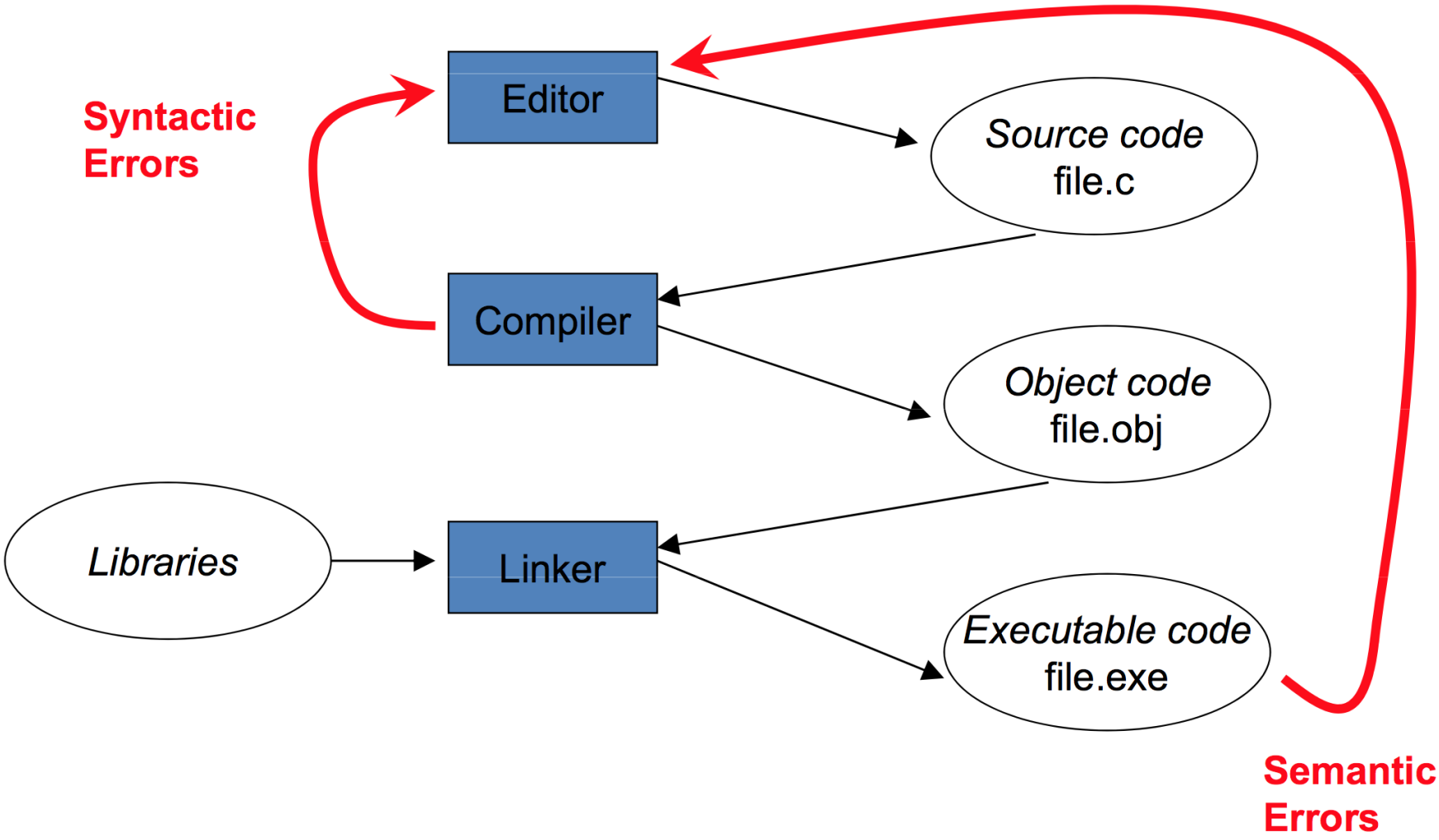
Το μορφότυπο (format) της C

- Οι εντολές τερματίζουν με ερωτηματικό ;
- Είναι καλή πρακτική να χρησιμοποιούμε indentation για να διαβάζεται το πρόγραμμα πιο εύκολα
- Ωστόσο το μορφότυπο είναι ελεύθερο (free format): τα κενά και η ενδοπαραγραφοποίηση δεν αναγνωρίζονται από τον compiler
- Η C είναι case sensitive – προσοχή στα πεζά/κεφαλαία !
  - Ολα τα C keywords και οι συναρτήσεις γράφονται με πεζα.
  - Γράφοντας INT, Int, κλπ αντί για int είναι σφάλμα
- Strings are placed in double quotes
- Η νέα γραμμή αντιπροσωπέυεται από \n (Escape sequence)

Διαδοχικές φάσεις ενός προγράμματος



Απασφαλμάτωση λαθών



Συντακτικά και σημασιολογικά λάθη

- **Syntax errors:** όταν χρησιμοποιούμε λανθασμένα τους κανόνες γραμματικής της γλώσσας (grammar)
  - "*Me not speak English good*."
  - Χρήση έγκυρων C συμβόλων σε λάθος μέρη
  - Ανιχνεύονται από τον μεταγλωτιστή
- **Semantics errors**: λάθη στο νόημα:
  - "*This sentence is excellent Italian*."
  - Σωστά συντακτικά προγράμματα που παράγουν λάθος έξοδο
  - Ο χρήστης παρατηρεί την έξοδο του προγράμματος

Δεύτερο πρόγραμμα

```
#include <stdio.h>
int main (void)
{
    printf ("Programming is fun.\n");
    printf ("And programming in C is even more fun.\n");
    return 0;
}
```

Εκτύπωση πολλαπλών γραμμών

```
#include <stdio.h>
int main (void)
{
   printf ("Testing...\n..1\n...2\n....3\n");
   return 0;
}
```

It is not necessary to make a separate call to printf for each line of output !

Output:

```
Testing...
..1
...2
....3
```

Μεταβλητές

- Τα προγράμματα μπορούν να χρησιμοποιήσουν συμβολικά ονόματα για να αποθηκεύσουν δεδομένα εισόδου και αποτελέσματα
- **Μεταβλητή**: ένα συμβολικό όνομα για μια θέση μνήμης
  - ο προγραμματιστής δε χρειάζεται να ανησυχεί (ή να γνωρίζει) την τιμή της διεύθυνσης μνήμης
- Στη C, οι μεταβλητές πρέπει να ***δηλωθούν*** πρωτού να χρησιμοποιηθούν

Μεταβλητές

- Η δήλωση μιας μεταβλητής δεσμεύει χώρο για να αποθηκευθεί η τιμή της μεταβλητής, χωρίς όμως να βάζει κάποια τιμή εκεί.
- **Ανάθεση**: οι τιμές μπαίνουν στη θέση μνήμης που έχει δεσμευθεί όταν γίνεται ανάθεση τιμής σε μια μεταβλητή (αρχικοποίηση της μεταβλητής)
- int a #δήλωση μεταβλητής
- int a, value1, value 2 # δήλωση τριών μεταβλητών
- x1=7 ; # ανάθεση τιμής

Μεταβλητές

- Κάποιες παλιότερες γλώσσες (Basic) αλλά και νεώτερες (python) δε δηλώνουν μεταβλητές.
- **Η δήλωση μεταβλητών είναι καλή ιδέα γιατί:**
  - Μπαίνουν όλες μαζί και δινουν μια εικόνα στο χρήστη
  - Αναγκάζουν τον προγραμματιστή να κάνει ένα πλάνο πριν γράψει τον κώδικα
  - Προφυλάσσουν από συντακτικά λάθη στα ονόματα
  - Ο μεταγλωττιστής ξέρει πόση μνήμη να δεσμεύσει στατικά
  - Και μπορεί να ελέγξει ότι οι λειτουργίες στις οποίες εμπλέκεται η μεταβλητή συμφωνούν με τον τύπο της

Κανόνες μεταβλητών

- Τα ονόματα των μεταβλητών λέγονται identifiers
- Έγκυρα ονόματα μεταβλητών:
- Αρχίζουν με γράμμα ή _ και ακολουθούνται από οποιοδήποτε συνδυασμό γραμμάτων, _, και αριθμών
  - πχ sum, pieceFlag, J5x7, Number_of_Moves
- Τα ειδικά ονόματα της C δε μπορούν να είναι ονόματα μεταβλητών
  - sum$value, 3spencer
  - ο προγραμματιστής δε χρειάζεται να ανησυχεί (ή να γνωρίζει) την τιμή της διεύθυνσης μνήμης

Κανόνες μεταβλητών

- sum, Sum και SUM είναι διαφορετικές μεταβλητές
- Τα ονόματα των μεταβλητών μπορεί να είναι όσο μεγάλα θέλετε αλλά μόνο οι πρώτοι 31 ή 63 χαρακτήρες θα ληφθούν υπ'όψη από το μεταγλωττιστή.
- Είναι καλή πρακτική να χρησιμοποιούμε ονόματα που σχετίζονται με τη σημασία της μεταβλητή

Χρήση και εκτύπωση μεταβλητών

```
#include <stdio.h>
int main (void)
{
        int sum;
        sum = 50 + 25;
        printf ("The sum of 50 and 25 is %i\n", sum);
        return 0;
}
```

Variable sum **declared** of type int

Variable sum **assigned** expression 50+25

Value of variable sum is **printed** in place of %i

The printf routine call has now 2 arguments: first argument a string containing also a format specifier (%i), that holds place for an integer value to be inserted here

Εκτύπωση πολλαπλών τιμών

```
#include <stdio.h>
int main (void)
{
  int value1, value2, sum;
  value1 = 50;
  value2 = 25;
  sum = value1 + value2;
  printf ("The sum of %i and %i is %i\n",value1, value2, sum);
  return 0;
}
```

The format string must contain as many placeholders as expressions to be printed

Χρήση σχολίων σε ένα πρόγραμμα

- **Σχόλια** χρησιμοποιούνται σε ένα πρόγραμμα για να το τεκμηριώσουν και να βελτιώσουν την αναγνωσιμότητά του
- Είναι χρήσιμα μόνο για το χρήστη – ο μεταγλωτιστής τα αγνοεί
- Τρόποι εισαγωγής σχολίων στη C:
  - Σχόλια σε περισσότερες από μια γραμμές:

    αρχή: /*, τέλος: */
  - Σχόλια στο τέλος γραμμής: αρχή: //

Παράδειγμα

```
/* This program adds two integer values
and displays the results */

#include <stdio.h>
int main (void)
{
    // Declare variables
    int value1, value2, sum;
    // Assign values and calculate their sum
    value1 = 50;
    value2 = 25;
    sum = value1 + value2;
    // Display the result
    printf ("The sum of %i and %i is %i\n",
                                    value1, value2, sum);
    return 0;
}
```

Παρουσίαση του περιεχομένου

Τα κεφάλαια του βιβλίου που προτείνεται, είναι :

❶**ΓΝΩΡΙΜΙΑ ΜΕ ΤΗ ΓΛΩΣΣΑ C**

(Βασικές έννοιες και η δομή της γλώσσας)

❷**ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ C**

(Οι μεταβλητές, οι σταθερές, οι τελεστές και οι εκφράσεις, εντολές εισόδου/εξόδου)

❸**ΕΝΤΟΛΕΣ ΕΛΕΓΧΟΥ**

(Περιγραφή των εντολών ελέγχου)

Το πιο σημαντικό κεφάλαιο για ένα προγραμματιστή

❹**ΣΥΝΑΡΤΗΣΕΙΣ**

(Περιγραφή των συναρτήσεων, functions)

Εδώ ανακαλύπτουμε τον πλούτο και την ευελιξία του προγραμματισμού με τη γλώσσα C

Παρουσίαση του περιεχομένου (συνέχεια)

5 ΔΕΙΚΤΕΣ ΚΑΙ ΠΙΝΑΚΕΣ

(Οι έννοιες των δεικτών (pointers) και εφαρμογές των πινάκων (arrays))

6 ΕΠΕΞΕΡΓΑΣΙΑ ΑΡΧΕΙΩΝ

(Περιγραφή και χρήση των αρχείων στη γλώσσα C)

7 ΔΟΜΕΣ ΚΑΙ ΕΝΩΣΕΙΣ

(Εισαγωγή στις δομές και τις ενώσεις)

ΕΦΑΡΜΟΓΕΣ της γλώσσας C