



Κεφάλαιο 2.1-2.4:

**Εισαγωγή στην C**

(Διαλέξεις 3-4)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ



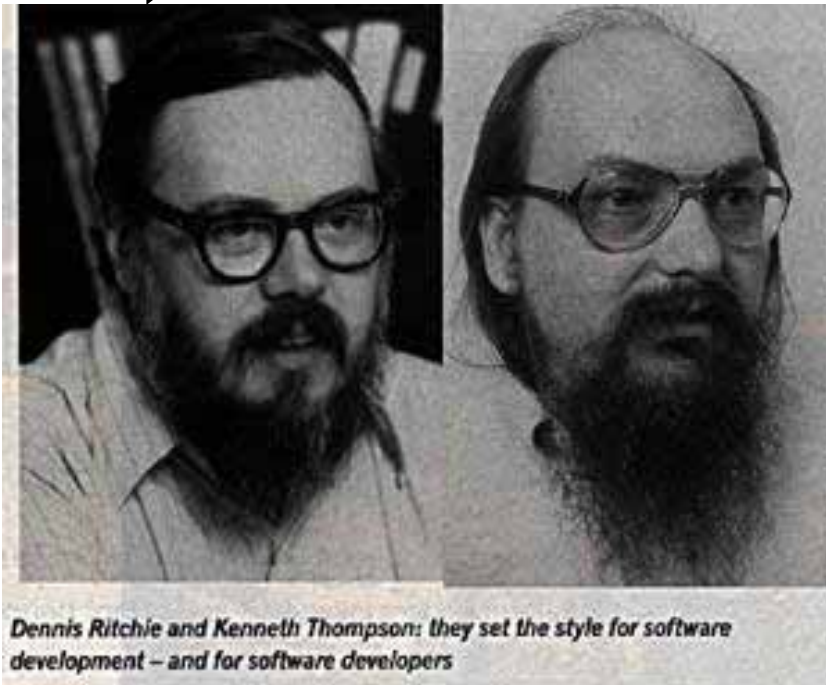
# Περιεχόμενα

- Εισαγωγή στην C:
- Σύνταξη και Σημασιολογία
  - σχόλια
  - μεταβλητές και σταθερές
  - τύποι δεδομένων
- Μετά αυτές τις δυο διαλέξεις θα μπορείτε να γράψετε και να κατανοήσετε απλά προγράμματα στην C!!!

# Οι σχεδιαστές της C



Dennis Ritchie



Η γλώσσα C αναπτύχθηκε το 1972 από τον Dennis Ritchie στα AT&T (Bell) Laboratories.

Dennis Ritchie



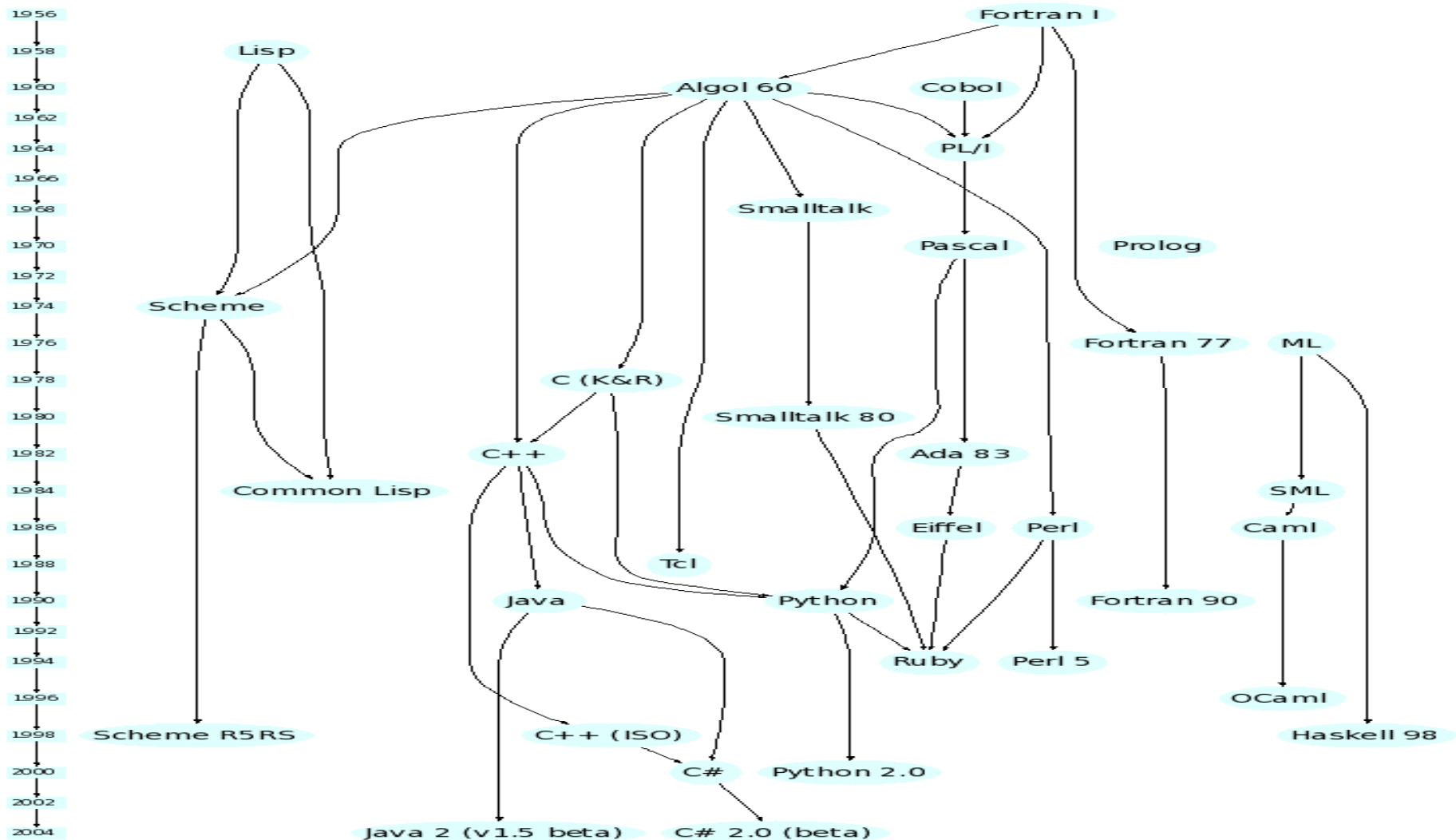
**ΕΠΛ 032: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕΘΟΔΩΝ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ**

# Η ιστορία της C



- Γλώσσα γενικού σκοπού
- Αρχικά μεγάλη συνάφεια με το UNIX
- Μεγάλο μέρος του UNIX γράφτηκε στη C
- Αλλά φυσικά και του Linux

# Ιστορικό Διάγραμμα Γλωσσών Προγραμματισμού



# Γιατί η C?

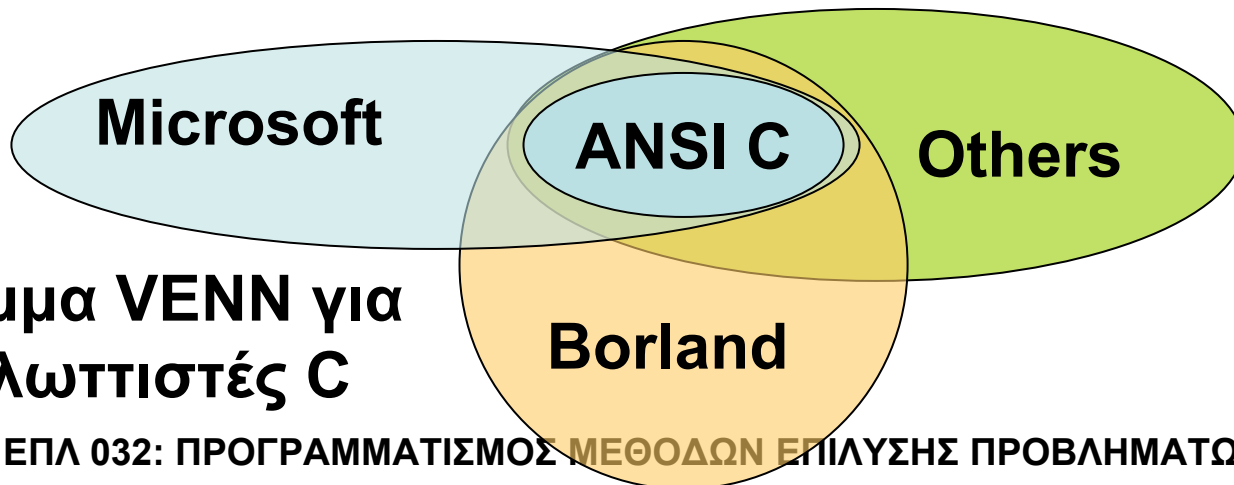


- C είναι μικρή με λίγες εντολές
- C είναι η γλώσσα του Unix/Linux
- C είναι standard
- C είναι φορητή
- C είναι ισχυρή
- C είναι αρθρωτή
- C είναι η βάση της C++ και της Java
- C είναι γρήγορη

# ANSI C Standard



- Η γλώσσα C είναι απλά ένα σύνολο από συντακτικούς κανόνες.
- Ξέρουμε όλοι ότι για να εκτελεστεί ένα πρόγραμμα C πρέπει να χρησιμοποιηθεί ένας **μεταγλώττισης (compiler)**
- Οι compilers παράγονται από διάφορες εταιρείες και οργανισμούς.
- Για να υπάρχει κάποιο πρότυπο, το **American National Standard Institutes (ANSI)** δημιούργησε το πρότυπο Ansi C το οποίο καλούνται οι διάφορες εταιρείες να ακολουθούν.



Διάγραμμα VENN για μεταγλωττιστές C

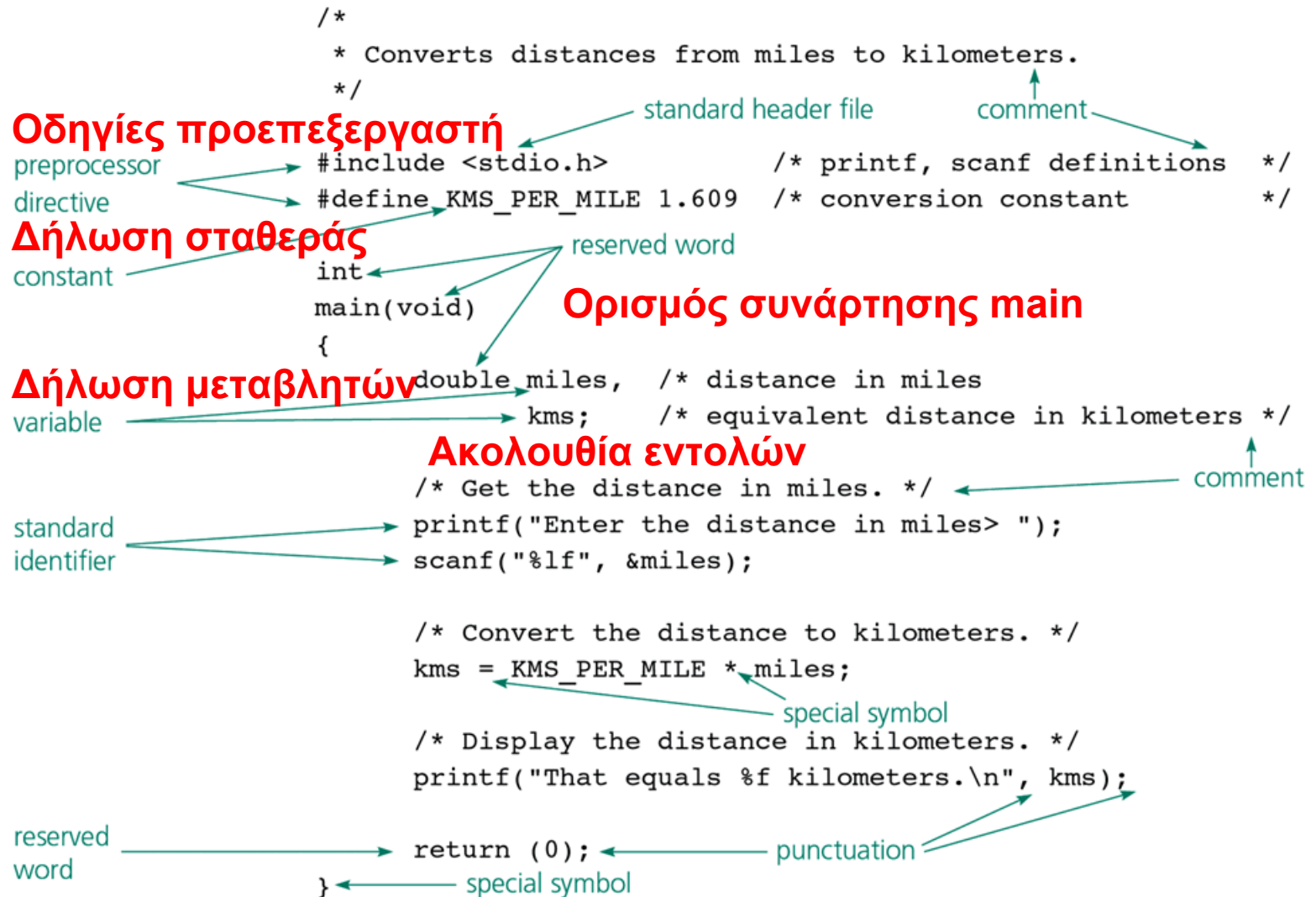


# Δομή Προγράμματος C



- A) Οδηγίες στον προεπεξεργαστή
- B) Δηλώσεις σταθερών
- Γ) Ορισμός συνάρτησης `main`
  - δηλώσεις μεταβλητών
  - εντολές (εκφράσεις, κλήσεις συναρτήσεων κτλ)
- Δ) Σχόλια

# Δομή προγράμματος C



# A) Οδηγίες προς τον προεπεξεργαστή



Εισδοχή Βιβλιοθηκών `#include`

Σύνταξη: `#include <standard header file>`

```
#include <stdio.h>
```

```
#include <math.h>
```

Μία βιβλιοθήκη (library) είναι μία συλλογή χρήσιμων συναρτήσεων και συμβόλων.

- Άλλες βιβλιοθηκες: [<time.h>](#) , **string.h**, **signal.h**, .....
- Οι πιο μοντέρνες γλώσσες (4ης γενιάς – Java) έχουν ένα τεράστιο σύνολο από βιβλιοθήκες. Αυτό τις καθιστά ουσιαστικά πιο ισχυρές για την επίλυση προβλημάτων
- Άλλες διαταγές στον προ-επεξεργαστή:
  - `define`, `ifdef`, `if`, `else`, `endif`, `undef`

# A) Οδηγίες προς τον προεπεξεργαστή



- Η βιβλιοθήκη **stdio** περιέχει, ανάμεσα σε άλλα, τις συναρτήσεις **printf**, για εκτύπωση πληροφοριών, και **scanf**, για εισδοχή πληροφοριών, καθώς επίσης όλη την λειτουργικότητα για ανάγνωση / γραφή πληροφοριών σε αρχεία.
- Η βιβλιοθήκη **math** περιέχει διάφορες μαθηματικές συναρτήσεις.

# B) Σταθερές, Μεταβλητές και Συναρτήσεις



- Ένα πρόγραμμα σε C, ανεξαρτήτως μεγέθους, αποτελείται από εντολές (οργανωμένα μέσα σε συναρτήσεις - functions) και δεδομένα (οργανωμένα μέσα σε μεταβλητές - variables).

Συνάρτηση → Σύνολο εντολών που προσδιορίζουν τις υπολογιστικές λειτουργίες που θα γίνουν

- Μεταβλητές → Αποθηκεύουν τιμές που χρησιμοποιούνται κατά τον υπολογισμό. Οι τιμές προέρχονται από α) το δίσκο β) κάποια μονάδα εισόδου (πχ keyboard) γ) παράγονται κατά την διάρκεια εκτέλεσης του προγράμματος

## B) Σταθερές VS Μεταβλητές



- **Μεταβλητή (variables)**: Μια περιοχή στην μνήμη (RAM) του υπολογιστή όπου μπορούμε να αποθηκεύσουμε προσωρινά δεδομένα. Κατά την διάρκεια εκτέλεσης αυτή η τιμή μπορεί να αλλάξει όσες φορές θέλουμε  
  
πχ ένας μετρητής, η ηλικία του χρήστη, κτλ
- **Σταθερά (constants)**: Μια περιοχή στην μνήμη (RAM) του υπολογιστή όπου μπορούμε να αποθηκεύσουμε δεδομένα τα οποία **δεν αλλάζουν** κατά την διάρκεια εκτέλεσης ενός προγράμματος.
- Πχ.  $KMS\_PER\_MILE = 1.609$  (Ένα μίλι είναι πάντοτε 1.609 χιλιόμετρα)

# Ονομασίες σε Προγράμματα



- Για σταθερές, μεταβλητές, συναρτήσεις, κτλ.
- **Κανόνες Σύνταξης:**
  1. Αποτελούνται από γράμματα, δεκαδικά ψηφία και underscores ( \_ ).
  2. Δεν μπορεί να αρχίζουν με **δεκαδικό ψηφίο**.
  3. σύμβολα όπως **&, #, \$** δεν επιτρέπονται.
  4. το όνομα μιας μεταβλητής δεν μπορεί να περιέχει **κενό**
  5. **Δεσμευμένες λέξεις** (θα εξηγηθούν αργότερα) δεν μπορούν να χρησιμοποιηθούν ως ονομασίες για κάτι άλλο.
  6. Ονομασίες που ορίζονται σε κάποια από τις βασικές βιβλιοθήκες δεν πρέπει να **ξαναορίζονται** (πχ printf())

## GOOD

KMS\_PER\_MILE  
miles  
kms

## BAD

1Letter  
one new  
new-fn

## RESERVED

double  
printf  
scanf

# Ονομασίες συνέχεια



- Η C είναι **CASE SENSITIVE**, δηλαδή κεφαλαία και μικρά γράμματα θεωρούνται διαφορετικά:
  - foo και FOO είναι δυο διαφορετικά ονόματα
  - One, one και ONE είναι όλες διαφορετικές μεταβλητές.
- Χρησιμοποιείτε αυτοεπεξηγηματικά ονόματα
  - Π.χ. η ονομασία **age** είναι πολύ καλύτερη από την ονομασία **A** για μία μεταβλητή στην οποία θα καταχωρούνται ηλικίες.



# Ονομασία Συνέχεια:

## Δεσμευμένες Λέξεις(Reserved Words)



- Λέξεις με ειδική σημασία για την C. Δεν πρέπει να ορίζονται ξανά σαν ονόματα

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

**Για περισσότερες Πληροφορίες:**

**<http://tigcc.ticalc.org/doc/keywords.html#auto>**

# Ορισμοί Σταθερών



- Σύνταξη: `#define` ονομασία τιμή

```
#define pi 3.1453  
#define false 0      #define true 1  
#define KMS_PER_MILE 1.609
```

- Η χρήση σταθερών
  - Αυξάνει την αναγνωσιμότητα του προγράμματος
  - Διευκολύνει τροποποιήσεις

# Σημασία Σταθεράς



- Σταθερές έχουν:
  - τιμή
  - όνομα **όχι όμως διεύθυνση**
  - Μεταγλωττιστής αντικαθιστά το όνομα με την τιμή

# Συνάρτηση main



- Κάθε πρόγραμμα έχει μία συνάρτηση με το όνομα main από όπου αρχίζει η εκτέλεση του προγράμματος.

Σύνταξη: void main (void) { σώμα συνάρτησης }

```
#include <stdio.h>
void main (void)
{
    int a;
    a = 5;
    printf("Hello %d\n",a);
}
```

**δήλωση**

**εντολή**

- Το σώμα μιας συναρτησης αποτελείται από δηλώσεις (declarations) και εντολές (executable statements).

# Μεταβλητές



- Αποθήκευση και ανάγνωση τιμών
- Κάθε μεταβλητή έχει:
  - τύπο
    - βασικοί: int, char, float, double
    - μέγεθος: 4B, 1B, 4B, 8B (κάθε κυψελίδα 1 Byte)
  - τιμή

Εξαρτάται από  
το λειτουργικό  
σύστημα

# Δηλώσεις Μεταβλητών



- Σύνταξη: τύπος λίστα-μεταβλητών ;

```
double miles;  
int count;  
double kms;
```

```
double miles, kms;  
int count = 0;  
/* αρχικοποίηση παράλληλα με δήλωση */
```

- Κάθε μεταβλητή που χρησιμοποιείται σε κάποιο πρόγραμμα χρειάζεται να **δηλωθεί**.
- Η δήλωσή της έχει ως αποτέλεσμα την παραχώρηση μνήμης για τη δημιουργία της.
- Ο τύπος (double, int, κτλ) της μεταβλητής προσδιορίζει τον απαιτούμενο χώρο μνήμης.

# Χρήση σχολίων σε C



- Σύνταξη: ξεκινούν με */\** και τελειώνουν με *\*/*  
ή *//* μας επιτρέπει να βάλουμε σχόλιο σε μια γραμμή
- Καμιά σημασία αλλά πολύ σημαντικό μέρος ενός προγράμματος (τεκμηρίωση)

*/\* auto einai ena syntaktika ortho sxolio\*/*

*/\* kai auto*

*\* einai*

*\* syntaktika*

*\* ortho*

*\*/* ΕΠΛ 032: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕΘΟΔΩΝ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ

# Χρήση σχολίων σε C



## Παράδειγμα 1

```
/*program example1.c
this program prints on your screen the expression
University of Cyprus*/

#include <stdio.h> // use of pre-processor
Int main ( )
{
    printf ("University of Cyprus\n");    /* printing UCY */
    return 0;
}
```



# Χρήση σχολίων σε C



## Παράδειγμα 2

```
/*  
** PROJECT: P1  
** FILE NAME: example1.c  
** AUTHOR: Demetris Zeinalipour  
** DATE: 19 September 2005  
**  
#include <stdio.h> /*use of pre-processor*/  
int main ( )  
{  
    printf ("University of Cyprus\n"); /*use of printf*/  
    return 0;  
}
```

# Για το μάθημα μας



```
/*  
** EPL032.1 Programmatismos Methodwn Epilyshs Provlimatwn  
** AUTHOR:          Demetris Zeinalipour  
** ID:              1234566  
** DATE:            20 January 2006  
** FILE NAME:       example1.c  
**  
/* this program prints on your screen the expression University of Cyprus */  
  
#include <stdio.h>    /*use of pre-processor*/  
int main ( )  
{  
    printf ("University of Cyprus\n");  
    return 0;  
}
```

# Χρήση σχολίων σε C



```
/* auto * ti* einai */
```

```
/******auto??? *****/
```

```
/* lathos sxolio */ giati */
```

```
*/ einai lathos??? /*
```

# Περίληψη Βασικών Σημείων Προγράμματος σε C



Προσοχή όλα με μικρά  
γράμματα (Lower case)

## Πρόγραμμα σε C

```
#include <stdio.h>

void main (void )
{
    printf ("University of Cyprus\n");
}
```

## Αποτέλεσμα

University of Cyprus

# Περίληψη Βασικών Σημείων Προγράμματος σε C



## 1. `#include <stdio.h>`

περιέχεται στην αρχή κάθε προγράμματος και περιλαμβάνει οδηγίες για τον προεπεξεργαστή

## 2. `main ( )`

Το σημείο έναρξης του προγράμματος είναι η «κύρια» συνάρτηση `main`.

## 3. `{ }`

δηλώνουν τον χώρο των εντολών, οι οποίες περιλαμβάνονται στη συνάρτηση `main` και πρέπει να εκτελεσθούν

# Περίληψη Βασικών Σημείων Προγράμματος σε C



## 4. `printf`

είναι μια συνάρτηση της βιβλιοθήκης της C και εμφανίζει στην οθόνη ότι δίνεται μέσα στην παρένθεση

## 5. `\n`

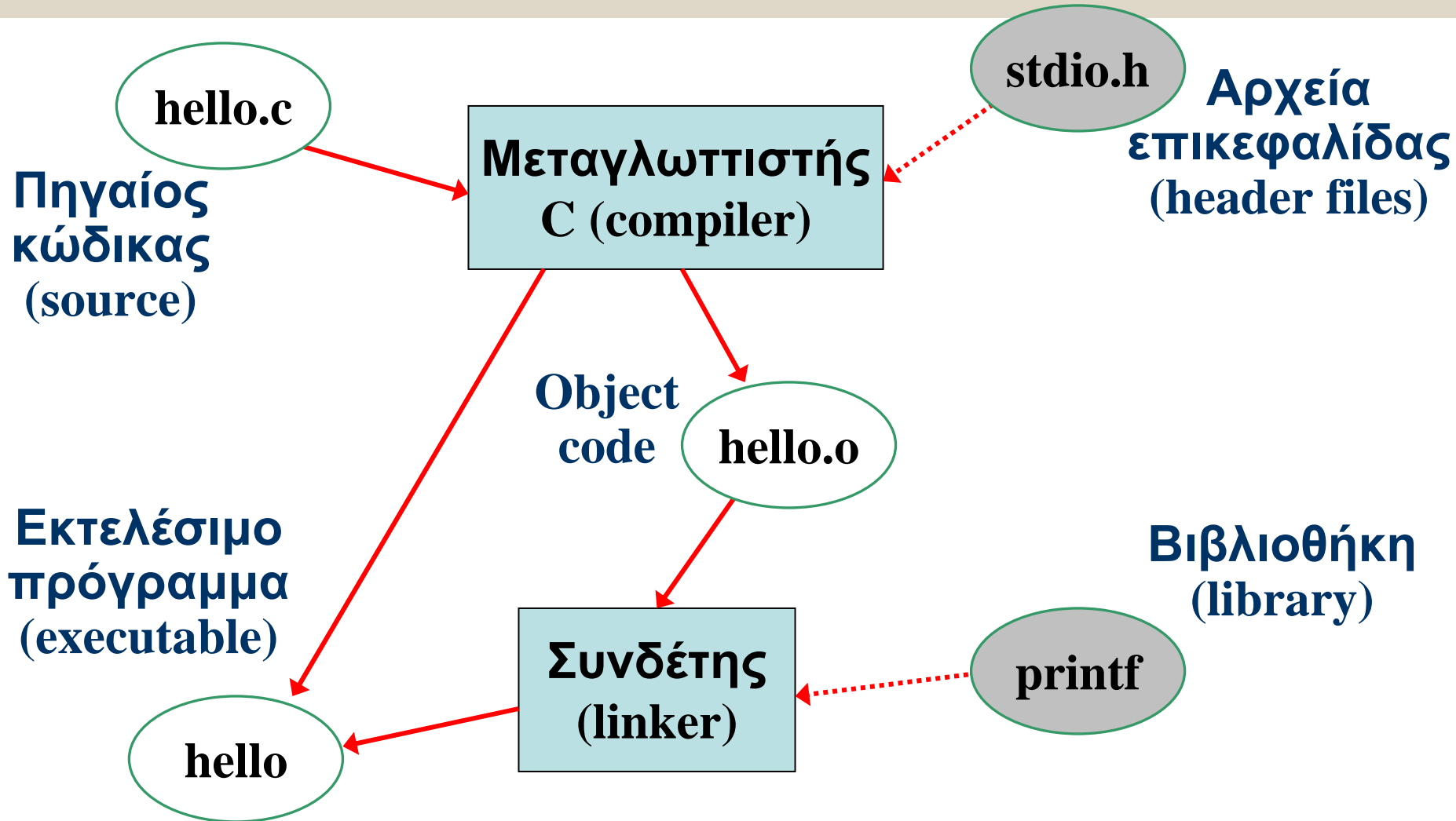
οδηγεί τον κέρσορα (cursor) να πάει στην επόμενη γραμμή

## 6. `;`

όλες οι **δηλώσεις** των C-προγραμμάτων καταλήγουν σε αυτό το σύμβολο (semicolon). Χρησιμοποιείται προκειμένου να φανεί που τελειώνει μια δήλωση

## 7. `/* */` Μεταξύ των συμβόλων αυτών γράφονται τα σχόλια.

# Περίληψη Βασικών Σημείων Προγράμματος σε C





# Παράδειγμα Προγραμματισμού

- Γράψετε ένα πρόγραμμα που μετατρέπει κυπριακές λίρες σε ευρώ.





# Κατανόηση και Αρχική Λύση

- Παράδειγμα: 10 λίρες, 2.12 ευρώ στην λίρα >>>>>> 21.2 ευρώ
- Δεδομένα (εισόδου): λίρες, rate
- Δεδομένα (εξόδου): ευρώ
- Υπολογισμός:

$$\text{ευρώ} = \text{λίρες} * \text{rate}$$



# Υλοποίηση

- με γλώσσα προγραμματισμού (C)
- επεξεργαστή κειμένου

```
/* programma pou metatrepei lires se euro */  
Int main()  
{  
    float lires, rate;           /* inputs*/  
    float euro;                 /* output*/  
  
    euro = lires * rate;  
    return 0;  
}
```

**Λείπει επικοινωνία με τον χρήστη**



```
/* programma pou metatrepei lires se euro */
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    float lires, rate;
```

```
    /* inputs*/
```

```
    float euro;
```

```
    /* output*/
```

```
    /* eisagwgh dedomenwn */
```

```
    printf("Enter lires kai rate: ");
```

```
    scanf("%f%f",&lires, &rate);
```

```
    euro = lires * rate;
```

```
    /*exagwgh dedomenwn*/
```

```
    printf("%.2f lires antistoixoun se %.2f euro\n", lires, euro);
```

```
    return 0;
```

```
}
```

# Ροή Ελέγχου



```
/* programma pou metatrepei lires se euro */
#include <stdio.h>
int main()
{
    float lires, rate;           /* inputs*/
    float euro;                 /* output*/

    /* eisagwgh dedomenwn */
    printf("Enter lires kai rate: ");
    scanf("%f%f",&lires, &rate);

    euro = lires * rate;

    /*exagwgh dedomenwn*/
    printf("%.2f lires antistoixoun se %.2f euro\n", lires, euro);

    return 0;
}
```

# Ροή Δεδομένων



```
/* programma pou metatrepei lires se euro */  
#include <stdio.h>  
int main()  
{  
    float lires, rate;           /* inputs*/  
    float euro;                 /* output*/  
  
    /* eisagwgh dedomenwn */  
    printf("Enter lires kai rate: ");  
    scanf("%f%f",&lires, &rate);  
  
    euro = lires * rate;  
  
    /*exagwgh dedomenwn*/  
    printf("%.2f lires antistoixoun se %.2f euro\n", lires, euro);  
  
    return 0;  
}
```

# Δοκιμή: Μεταγλωτισμός, Εκτέλεση



```
> gcc exchange.c -o exchange
```

```
> exchange
```

```
Enter lires kai rate: 100 1.78
```

```
100.00 lires antistoixoun se 178.00 euro
```

**Κατακρίβειαν  
8.94487**

```
> exchange
```

```
Enter lires kai rate: 5.5 1.62634
```

```
5.50 lires antistoixoun se 8.95 euro
```

**αλλά επειδή ζητήσαμε  
μόνο “.2f” για αυτό  
γίνεται rounding στο  
κοντινότερο δεκαδικό**

```
>
```

# Ροή Έλεγχου (control flow)



- Η ροή ελέγχου στην C είναι ακολουθιακής μορφής - sequential
  - Ξεκινά από την πρώτη εντολή της main
  - Εντολές εκτελούνται σε σειρά
- Υπάρχουν γλώσσες μή-ακολουθιακές αλλά δεν θα μας απασχολήσουν σε αυτό το μάθημα.

# Σημασία Μεταβλητής

