



# Πανεπιστήμιο Πειραιώς

## Τμήμα Ψηφιακών Συστημάτων

# Γλώσσα Προγραμματισμού C

## 2<sup>η</sup> Διάλεξη

Δημοσθένης Κυριαζής

Παρασκευή 12 Οκτωβρίου 2018



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Δεσμευμένες λέξεις
- Μεταβλητές
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While



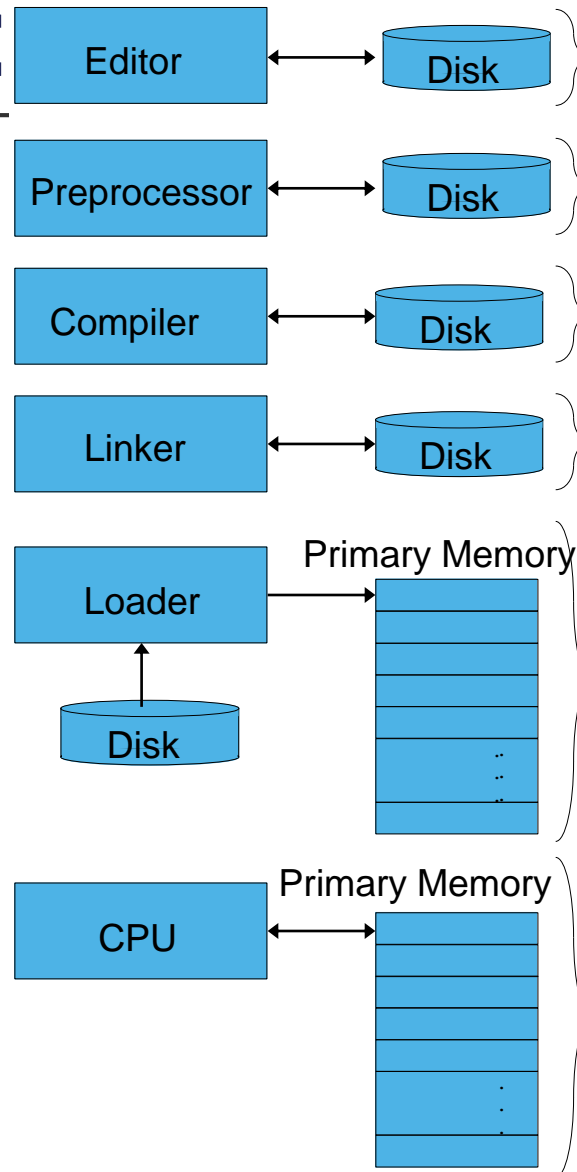
# Προγραμματισμός

---

- Ανάπτυξη αλγορίθμων και υλοποίηση τους σε γλώσσες προγραμματισμού για την επίλυση προβλημάτων
- Βήματα
  - Ορισμός προβλήματος
  - Ανάλυση προβλήματος
    - Δεδομένα, αποτελέσματα, εξαγόμενα
    - Τι πρέπει να γίνει
  - Υλοποίηση αλγόριθμου (σύνταξη προγράμματος)
  - Δοκιμή και επαλήθευση
    - Αποσφαλμάτωση (debugging)
  - Συντήρηση
    - Διορθωτική
    - Επαυξητική
      - Τελειοποίηση
      - Προσαρμογή

# Φάσεις ενός προγράμματος C

- Σύνταξη κώδικα (Edit)
- Προ-επεξεργασία (Preprocess)
- Μεταγλώττιση (Compile)
- Σύνδεση (Link)
- Φόρτωση Load
- Εκτέλεση (Execute)



Το πρόγραμμα γράφεται σε έναν συντάκτη κειμένου και αποθηκεύεται στο δίσκο.

Το πρόγραμμα προ-επεξεργαστή επεξεργάζεται τον κώδικα

Ο μεταγλωττιστής μεταγλωττίζει το πρόγραμμα σε κώδικα γλώσσας μηχανής δημιουργεί και αποθηκεύει το αποτέλεσμα στο δίσκο

Ο συνδέτης συνδέει τον κώδικα που παράγεται από το μεταγλωττιστή με τις βιβλιοθήκες

Το πρόγραμμα φορτώνεται στη μνήμη

Η CPU παίρνει κάθε εντολή και την εκτελεί, πιθανώς αποθηκεύοντας νέες τιμές δεδομένων καθώς εκτελείται το πρόγραμμα

# Ένα απλό πρόγραμμα σε C: Εκτύπωση μηνύματος στην οθόνη



```
1  /* simple.c
2     A first program in C */
3  #include <stdio.h>
4
5  /* function main begins program execution */
6  int main (void)
7  {
8     printf( "welcome to C!\n" );
9
10     return 0; /* indicate that program ended successfully */
11
12 } /* end function main */
```

```
Welcome to C!
```

## □ Σχόλια

- Κείμενο που περιέχεται μεταξύ /\* και \*/ (αγνοείται από τον υπολογιστή)
- Χρησιμοποιούνται για να περιγράψουν το πρόγραμμα



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Δεσμευμένες λέξεις
- Μεταβλητές
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While



# Δεσμευμένες λέξεις

## □ ANSI C

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

- Περιορισμός για μια γλώσσα, αλλά αυξάνουν την αναγνωσιμότητα και αξιοπιστία των προγραμμάτων και επιταχύνουν τη διαδικασία της μεταγλώττισης
- Πολλές είναι αποδεκτές σε πολλές γλώσσες προγραμματισμού

# Γενική μορφή προγράμματος στη C



εντολές προεπεξεργαστή (`#include`, `#define`, ...)

δηλώσεις μεταβλητών

δηλώσεις συναρτήσεων

```
int main (void)
```

```
{
```

```
εντολές
```

```
}
```

```
τύπος επιστρεφόμενης-τιμής f1(λίστα παραμέτρων)
```

```
{ ... }
```

```
τύπος επιστρεφόμενης-τιμής f2 (λίστα παραμέτρων)
```

```
{ ... }
```

```
⋮
```

```
τύπος επιστρεφόμενης-τιμής fN (λίστα παραμέτρων)
```

```
{ ... }
```



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Δεσμευμένες λέξεις
- Μεταβλητές
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While



# Μεταβλητές (1/2)

---

- Μια μεταβλητή αντιστοιχεί σε μια θέση της μνήμης που έχει το όνομα της μεταβλητής
- Κάθε μεταβλητή
  - όνομα (αναγνωριστικό)
  - τύπος
  - μέγεθος
  - τιμή
- Η δήλωση μιας μεταβλητής προσδιορίζει στο μεταγλωττιστή τον τύπο των δεδομένων που χρησιμοποιεί η μεταβλητή
  - τύπος όνομα-μεταβλητής



# Μεταβλητές (2/2)

---

- Κανόνες ονόματος ή αναγνωριστικού
  - Πρέπει να αρχίζει υποχρεωτικά με γράμμα ή με \_
  - Μπορεί να περιέχει γράμματα, αριθμούς και χαρακτήρες υπογράμμισης (\_)
  - Δεν μπορεί να είναι δεσμευμένη λέξη
  - Μικροί και κεφαλαίοι χαρακτήρες θεωρούνται διαφορετικοί
    - π.χ. `count` και `COUNT` θεωρούνται διαφορετικές μεταβλητές
- Η C υποστηρίζει πέντε βασικούς τύπους δεδομένων



# Βασικοί τύποι δεδομένων της C

---

## □ Τύπος

χαρακτήρας

ακέραιος

αριθμός κινητής υποδιαστολής

αριθμός κινητής υποδιαστολής, διπλής ακρίβειας

απουσία τιμής

## Δεσμευμένη λέξη

char

int

float

double

void

## □ Παραδείγματα

```
char ch;
```

```
int counter;
```

```
float f;
```

```
double d;
```

```
float x, y, z;
```



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Δεσμευμένες λέξεις
- Μεταβλητές
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While



# Τελεστές (1/5)

---

## □ Κατηγορίες τελεστών

Αριθμητικοί

+ - \* / %

Σχεσιακοί

> >= == < <= !=

Λογικοί

&& || !

Διαχείρισης μνήμης

& [] . ->

Διαχείρισης των bits ενός byte

>> & | ^ ~



# Τελεστές (2/5)

## □ Αριθμητικοί τελεστές

### ■ Δυαδικοί $+$ $-$ $*$ $/$ $\%$ (τελεστής υπολοίπου)

x	y	x/y	x%y
4	2	2	0
4	3	1	1

### ■ Μοναδιαίοι

□ Μοναδιαίοι αύξησης και μείωσης  $++$   $--$

□ Προπορευόμενοι, π.χ.  $++X$

- η αύξηση ή η ελάττωση γίνεται πριν να χρησιμοποιηθεί η τιμή της μεταβλητής στον υπολογισμό της τιμής της έκφρασης στην οποία συμμετέχει

□ Παρελκόμενοι, π.χ.  $X++$

- η αύξηση ή η ελάττωση γίνεται μετά τον υπολογισμό της τιμής της έκφρασης στην οποία συμμετέχει



# Τελεστές (3/5)

- Αριθμητικοί τελεστές αύξησης και μείωσης

Εντολή	Τιμή x	Τιμή y
<code>int x=10, y=20;</code>	10	20
<code>++x;</code>	11	20
<code>y= --x;</code>	10	10
<code>y = x-- + y;</code>	9	20
<code>y = y - x++;</code>	10	11



# Τελεστές (4/5)

---

## □ Σχεσιακοί τελεστές

### ■ Τελεστής

>

### Ενέργεια

Μεγαλύτερο από

>=

Μεγαλύτερο ή ίσο από

<

Μικρότερο από

<=

Μικρότερο ή ίσο από

==

Ίσο

!=

Διάφορο



# Τελεστές (5/5)

---

## □ Λογικοί τελεστές

### ■ Τελεστής

&&

||

!

### Ενέργεια

AND (λογική σύζευξη)

OR (λογική διάζευξη)

NOT (λογική άρνηση)



# Εκφράσεις (1/2)

- Γενική περίπτωση
  - Αποτελείται από έναν ή περισσότερους τελεστέους και προαιρετικά από έναν ή περισσότερους τελεστές
- Τελεστέοι: μεταβλητές, σταθερές ή κλήσεις συναρτήσεων
- Κάθε έκφραση έχει μια τιμή
- Παραδείγματα απλών εκφράσεων
  - 8
  - count
  - func(x)
  - MAX\_NUM
- Παραδείγματα σύνθετων εκφράσεων
  - 8 \* 24
  - count + 4
  - func(x) / 3
  - (count + 32) / MAX\_NUM



# Προτεραιότητα (1/3)

---

## □ Αριθμητικοί τελεστές

### ■ Καθορισμός επιπέδων προτεραιότητας για τους τελεστές

- \* και / στο ίδιο επίπεδο
- + και - στο ίδιο επίπεδο
- \* και / σε υψηλότερο επίπεδο από τους δυαδικούς + και -
- μοναδιαίοι +, . υψηλότερη προτεραιότητα
- \*, /, % ενδιάμεση προτεραιότητα
- δυαδικοί +, . χαμηλότερη προτεραιότητα



# Προτεραιότητα (2/3)

---

- Προσεταιριστικότητα (associativity)
  - Καθορισμός σειράς εφαρμογής περισσότερων τελεστών στο ίδιο επίπεδο (πολλά στιγμιότυπα του τελεστή)
    - $+$ ,  $-$ ,  $*$  και  $/$  είναι αριστερής προσεταιριστικότητας (εφαρμογή από αριστερά προς τα δεξιά)
    - Τελεστής ανάθεσης είναι **δεξιάς προσεταιριστικότητας !!!**
      - `num_1 = num_2 = 5`



# Προτεραιότητα (3/3)

---

## □ Σχεσιακοί και λογικοί τελεστές

Υψηλότερη

!

> >= < <=

== !=

&&

Χαμηλότερη

||



# Εκφράσεις (2/2)

- Χρήση παρενθέσεων για προσδιορισμό σειράς εφαρμογής τελεστών ή/και αύξηση αναγνωσιμότητας μιας έκφρασης
  - Πρώτα υπολογίζονται οι εσωτερικές παρενθέσεις
  - Προσοχή: Χρησιμοποιήστε παρενθέσεις όπου υπάρχει αβεβαιότητα

- Παραδείγματα

$$2 - 3 * 4$$

$$(2 - 3) * 4$$

$$2 - (3 * 4)$$

$$x + y + z * w$$

$$(x + (y + (z * w)))$$

$$2 + ((4 + 2) / (7 - 5)) - 6$$



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Δεσμευμένες λέξεις
- Μεταβλητές
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While



# Συναρτήσεις (1/2)

- Σύνολο εντολών που «προσφέρουν» κάποια λειτουργικότητα
- Μορφή συνάρτησης
  - τύπος επιστρεφόμενης-τιμής όνομα συνάρτησης (λίστα παραμέτρων)
  - Η επιστρεφόμενη τιμή αναφέρεται στην έξοδο της συνάρτησης και μπορεί να είναι
    - void: Δεν επιστρέφεται κάτι
    - char, int, float, double: Ένας από τους βασικούς τύπους δεδομένων
  - Η λίστα παραμέτρων αναφέρεται στα ορίσματα / είσοδο της συνάρτησης και μπορεί να είναι
    - void ή (): Δεν λαμβάνει είσοδο
    - Ένας / περισσότεροι από τους βασικούς τύπους με τις σχετικές μεταβλητές
  - Παράδειγμα: `int function1 (int x, char y, int z)`



## Συναρτήσεις (2/2)

---

- Η κλήση των συναρτήσεων γίνεται με το όνομά τους και τα σχετικά ορίσματα

```
int function1 ()
{
    int x;
    x = function2 (3, 4);
    printf("The second function result is %d\n", x);
}
int function2 (int x, int y)
{
    int result;
    result = 0;
    result = x + y;
    return result;
}
```



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Δεσμευμένες λέξεις
- Μεταβλητές
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While

# Γενικές και τοπικές μεταβλητές (1/2)



- Γενικές μεταβλητές (Global variables)
  - Δηλώνονται έξω από όλες τις συναρτήσεις
  - Μπορούν να προσπελάζονται από οποιαδήποτε συνάρτηση του προγράμματος
  
- Τοπικές μεταβλητές (Local variables)
  - Δηλώνονται μέσα σε μια συνάρτηση
  - Προσπελάζονται και είναι γνωστές μόνο μέσα στη συνάρτηση στην οποία δηλώνονται
  - Δημιουργούνται όταν καλείται η συνάρτηση και παύουν να υπάρχουν μετά τον τερματισμό της συνάρτησης
  - Οι τοπικές μεταβλητές μιας συνάρτησης δεν συνδέονται με τις τοπικές μεταβλητές μιας άλλης συνάρτησης

# Γενικές και τοπικές μεταβλητές (2/2)



εντολές προεπεξεργαστή (#include, #define, ...)

δηλώσεις γενικών μεταβλητών

δηλώσεις συναρτήσεων

```
int main (void)
```

```
{
```

```
εντολές
```

```
}
```

```
τύπος επιστρεφόμενης-τιμής f1(λίστα παραμέτρων)
```

```
{
```

δηλώσεις τοπικών μεταβλητών

```
... }
```

```
:
```

```
τύπος επιστρεφόμενης-τιμής fN (λίστα παραμέτρων)
```

```
{ ... }
```



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While

# Παράδειγμα πρόσθεσης 2 ακεραίων (1/5)



```
1  /* calc program */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int integer1; /* first number to be input by user */
8      int integer2; /* second number to be input by user */
9      int sum;      /* variable in which sum will be stored */
10
11     printf( "Enter first integer\n" ); /* prompt */
12     scanf( "%d", &integer1 ); /* read an integer */
13
14     printf( "Enter second integer\n" ); /* prompt */
15     scanf( "%d", &integer2 ); /* read an integer */
16
17     sum = integer1 + integer2; /* assign total to sum */
18
19     printf( "sum is %d\n", sum ); /* print sum */
20
21     return 0; /* indicate that program ended successfully */
22
23 } /* end function main */
```

# Παράδειγμα πρόσθεσης 2 ακεραίων (2/5)

---



## □ Αποτέλεσμα

```
Enter first integer
45
Enter second integer
72
Sum is 117
```

# Παράδειγμα πρόσθεσης 2 ακεραίων (3/5)



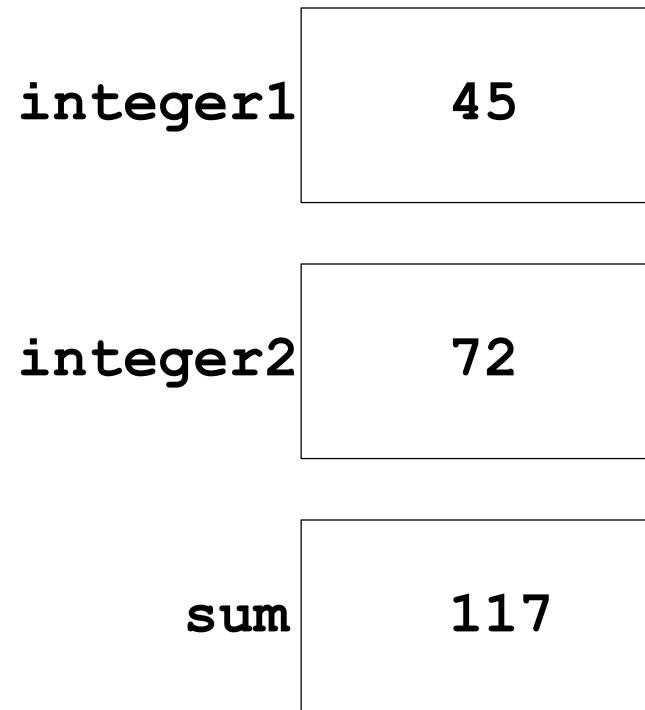
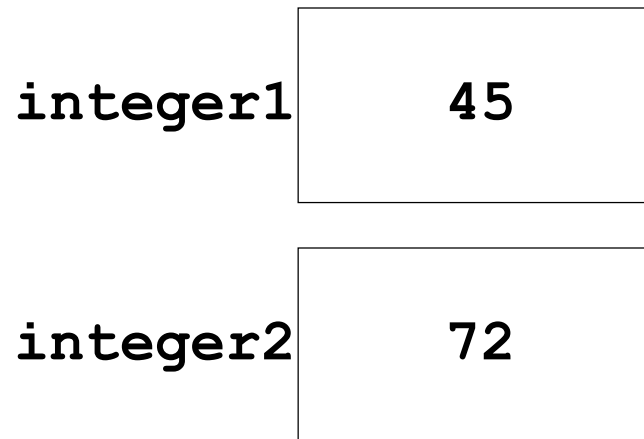
- Σχόλια, `#include <stdio.h>` και `main`
- `int integer1, integer2, sum;`
  - Δήλωση ακέραιων μεταβλητών
- `scanf( "%d", &integer1 );`
  - Συνάρτηση βιβλιοθήκης
  - χρησιμοποιείται για το διάβασμα μιας τιμής από το πληκτρολόγιο
  - Η συγκεκριμένη εντολή `scanf` έχει δύο ορίσματα
    - `%d` - υποδεικνύει πως τα δεδομένο εισόδου είναι ένας ακέραιος αριθμός
    - `&integer1` - υποδεικνύει τη θέση της μνήμης στην οποία πρέπει να αποθηκεύει η συγκεκριμένη τιμή
    - `&` - τελεστής μνήμης

# Παράδειγμα πρόσθεσης 2 ακεραίων (4/5)



- = τελεστής εκχώρησης (assignment operator)
  - Εκχωρεί μια τιμή σε μια μεταβλητή
    - `sum = variable1 + variable2;`
  - + τελεστής αθροίσματος
    - Η `sum` παίρνει την τιμή του αθροίσματος `variable1 + variable2;`
- `printf( "Sum is %d\n", sum );`
  - Ανάλογη με την `scanf`
    - `%d` σημαίνει ότι θα τυπωθεί ένας δεκαδικός ακέραιος
    - `sum` προσδιορίζει ποιος είναι ο ακέραιος που θα τυπωθεί
  - Πράξεις μπορούν να εκτελούνται εντός μιας εντολής `printf`
    - `printf( "Sum is %d\n", integer1 + integer2 );`

# Παράδειγμα πρόσθεσης 2 ακεραίων (5/5)





# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While

# Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού

---



- Κατανόηση βασικών τεχνικών επίλυσης προβλημάτων
- Ανάπτυξη αλγορίθμων με σχεδιασμό “από πάνω προς τα κάτω” (top-down)
- Χρήση εντολών if και if...else για την επιλογή ενεργειών
- Χρήση εντολής while για την επαναληπτική εκτέλεση εντολών σε ένα πρόγραμμα
- Χρήση ένθετων εντολών
- Κατανόηση επανάληψης ελεγχόμενης από μετρητή και ελεγχόμενης από τιμή «σημαία» / «φρουρό»



# Εισαγωγή

---

- Πριν γράψουμε ένα πρόγραμμα
  - Πρέπει να έχουμε κατανοήσει πολύ καλά το πρόβλημα που θέλουμε να λύσουμε
  - Σχεδιάζουμε προσεκτικά τον τρόπο επίλυσης του προβλήματος
- Ενώ γράφουμε ένα πρόγραμμα
  - Πρέπει να ξέρουμε ποιες δομικές μονάδες (“building blocks”) είναι διαθέσιμες
  - Πρέπει να χρησιμοποιούμε καλές αρχές-κανόνες προγραμματισμού



# Αλγόριθμοι

---

- Επίλυση προβλημάτων
  - Όλα τα προβλήματα μπορούν να επιλυθούν με την εκτέλεση μιας σειράς ενεργειών με συγκεκριμένη σειρά
- Αλγόριθμος: ορίζεται ως μια διαδικασία από
  - Ενέργειες που πρέπει να εκτελεστούν
  - Τη σειρά με την οποία εκτελούνται αυτές οι ενέργειες
- Έλεγχος προγράμματος
  - Ορισμός της σειράς με την οποία εκτελούνται οι εντολές



# Ψευδοκώδικας

---

- Τεχνητή γλώσσα που μας βοηθάει στην ανάπτυξη αλγορίθμων
- Μοιάζει με τη φυσική γλώσσα
- Δεν “εκτελείται” πραγματικά στους υπολογιστές
- Μας βοηθάει να σκεφτούμε ένα πρόγραμμα προτού το γράψουμε
  - Εύκολα το μετατρέπουμε στο αντίστοιχο C πρόγραμμα
  - Αποτελείται μόνο από εκτελέσιμες εντολές



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While



# Δομές ελέγχου

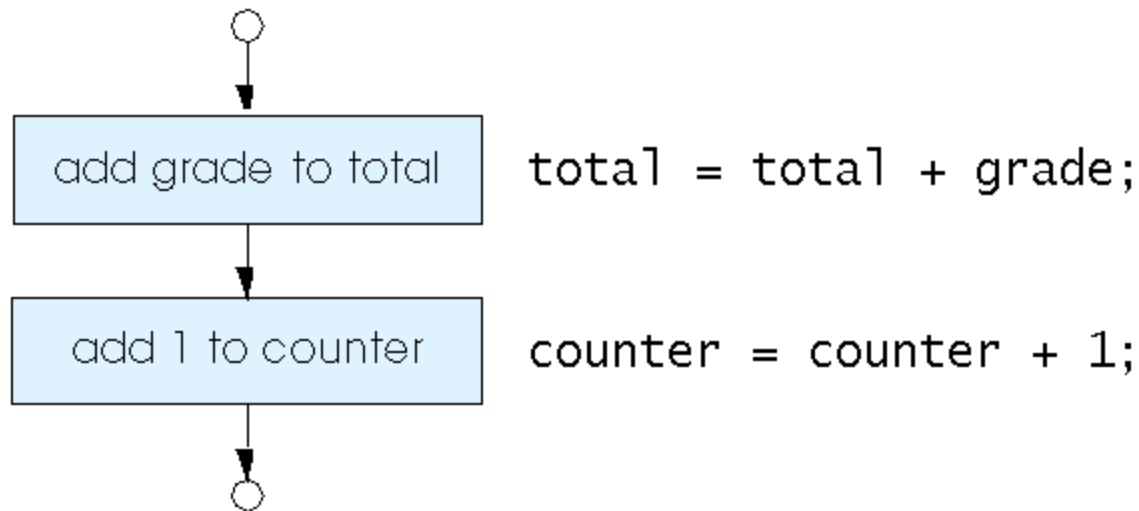
---

- Σειριακή εκτέλεση
  - Οι εντολές εκτελούνται η μια μετά την άλλη με τη σειρά που είναι γραμμένες
- Μεταφορά ελέγχου
  - Η επόμενη εντολή που εκτελείται δεν είναι η επόμενη στη σειρά
  - Η υπερβολική χρήση εντολών goto οδηγούσε σε πολλά προβλήματα
- Όλα τα προγράμματα γράφονται με βάση 3 δομές ελέγχου
  - Σειριακές δομές
    - Ενσωματωμένες στη C
      - Τα προγράμματα εκτελούνται σειριακά
  - Δομές επιλογής
    - Στη C υπάρχουν τρεις τύποι
      - if, if...else switch
  - Δομές επανάληψης
    - Στη C υπάρχουν τρεις τύποι
      - while, do...while και for



# Δομές ελέγχου

- Διάγραμμα ροής σειριακής δομής της C





# Δομές ελέγχου

---

- Διάγραμμα ροής (flowcharts)
  - Γραφική αναπαράσταση αλγόριθμου
  - Σχεδιάζεται χρησιμοποιώντας ορισμένα ειδικά σύμβολα που συνδέονται με βέλη που ονομάζονται γραμμές ροής (flowlines)
  - Ορθογώνιο παραλληλόγραμμο (σύμβολο ενέργειας)
    - Υποδηλώνει οποιονδήποτε τύπο ενέργειας
  - Έλλειψη
    - Υποδηλώνει την αρχή ή το τέλος ενός προγράμματος ή ενός τμήματος κώδικα
  
- Δομές ελέγχου μονής εισόδου / μονής εξόδου
  - Σύνδεση σημείου εξόδου μιας δομής με το σημείο εισόδου της επόμενης δομής (στοίβα δομών ελέγχου)
  - Διευκολύνει τη δημιουργία (το γράψιμο) προγραμμάτων



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While



# Η εντολή επιλογής if (1/4)

- Δομή επιλογής
  - Χρησιμοποιείται για την επιλογή μεταξύ εναλλακτικών ενεργειών
  - Σε ψευδοκώδικα
    - If ο βαθμός ενός φοιτητή είναι μεγαλύτερος ή ίσος του 5 τύπωσε "Passed"*
- Αν η συνθήκη της If είναι αληθής - true
  - Εκτελείται η εντολή τύπωσε και το πρόγραμμα συνεχίζει με την επόμενη εντολή
- Αν η συνθήκη της If είναι ψευδής - false
  - Η εντολή τύπωσε αγνοείται και το πρόγραμμα συνεχίζει με την επόμενη εντολή
- Προκειμένου να είναι πιο ευανάγνωστα τα προγράμματα
  - Χωρίζουμε τις διάφορες ενότητες τοποθετώντας τις αντίστοιχες παραγράφους λίγο δεξιότερα
    - Η C αγνοεί τα «κενα»



# Η εντολή επιλογής `if` (2/4)

## □ Εντολή ψευδοκώδικα σε C

```
if ( grade >= 5 )  
    printf( "Passed\n" );
```

- Ο κώδικας C μοιάζει αρκετά με τον ψευδοκώδικα

## □ Γενική μορφή εντολής `if`

```
if (έκφραση)  
    εντολή;
```

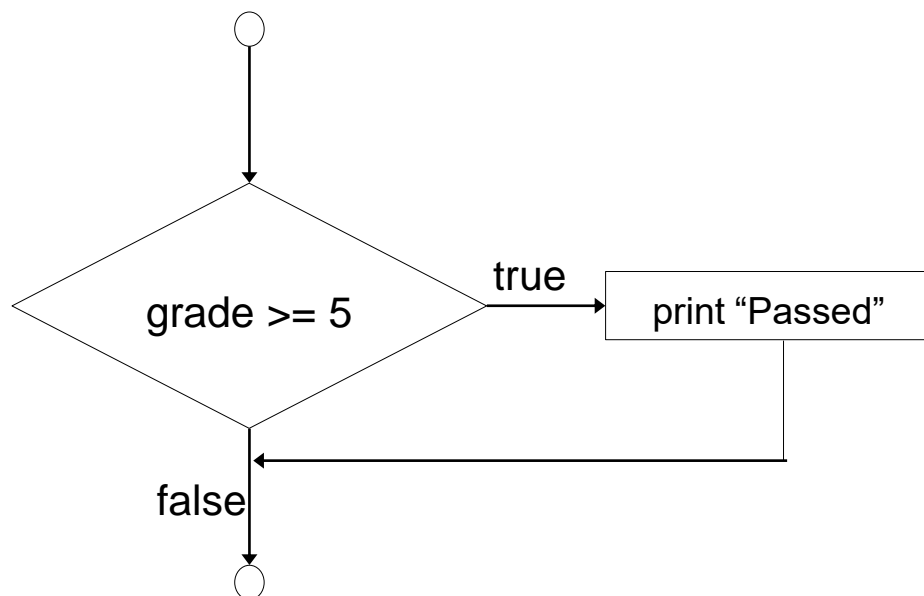
## □ Ρόμβος (σύμβολο απόφασης)

- Υποδεικνύει ότι πρέπει να παρθεί μια απόφαση
- Περιέχει μια έκφραση που μπορεί να είναι `true` (αληθής) ή `false` (ψευδής)
- Έλεγχος της συνθήκης, επιλογή κατάλληλης διαδρομής/ ροής



# Η εντολή επιλογής `if` (3/4)

- Η εντολή `if` είναι μια δομή μονής εισόδου / μονής εξόδου



Μια απόφαση μπορεί να γίνει σε κάθε έκφραση.  
0 - false  
μη μηδενική τιμή - true



# Η εντολή επιλογής `if` (4/4)

---

- Παράδειγμα: Γράψτε ένα πρόγραμμα το οποίο διαβάζει έναν αριθμό από το πληκτρολόγιο και εξακριβώνει αν αυτός ο αριθμός είναι αρνητικός ή μη-αρνητικός.



# Πρόγραμμα C

```
1  /*Decide if input is negative or non-negative */
2  #include <stdio.h>
3
4  /* function main begins program execution */
5  int main()
6  {
7      int num; /* number to be entered */
8
9      printf("Enter an integer: \n");
10     scanf("%d",&num);
11
12     if (num < 0)
13         printf("Number is negative.\n");
14     if (num >= 0)
15         printf("Number is non-negative.\n");
16
17     return 0;
22 } /* end main*/
```



# Αποτέλεσμα προγράμματος

---

```
Enter an integer:  
10  
Number is non-negative.  
Enter an integer:  
-78  
Number is negative.  
Enter an integer:  
0  
Number is non-negative.
```



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While



# Η εντολή επιλογής `if...else` (1/10)

---

## □ `if`

- Μια ενέργεια πραγματοποιείται μόνο αν η συνθήκη της `if` είναι `true`

## □ `if...else`

- Προσδιορίζει μια ενέργεια που πρέπει να εκτελεστεί τόσο όταν η συνθήκη είναι `true` όσο και όταν η συνθήκη είναι `false`

## □ Ψευδοκώδικας:

*If ο βαθμός ενός φοιτητή είναι μεγαλύτερος ή ίσος του 5*

*Τύπωσε "Passed"*

*else*

*Τύπωσε "Failed"*



# Η εντολή επιλογής `if...else` (2/10)

---

## □ Κώδικας C

```
if ( grade >= 5 )  
    printf( "Passed\n");  
else  
    printf( "Failed\n");
```

## □ Γενική μορφή εντολής `if...else`

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



# Η εντολή επιλογής `if...else` (3/10)

---

## □ Τριαδικός τελεστής (`?:`)

### ■ Έχει τρία ορίσματα

- συνθήκη ? εντολή if true : εντολή if false

## □ Παράδειγμα

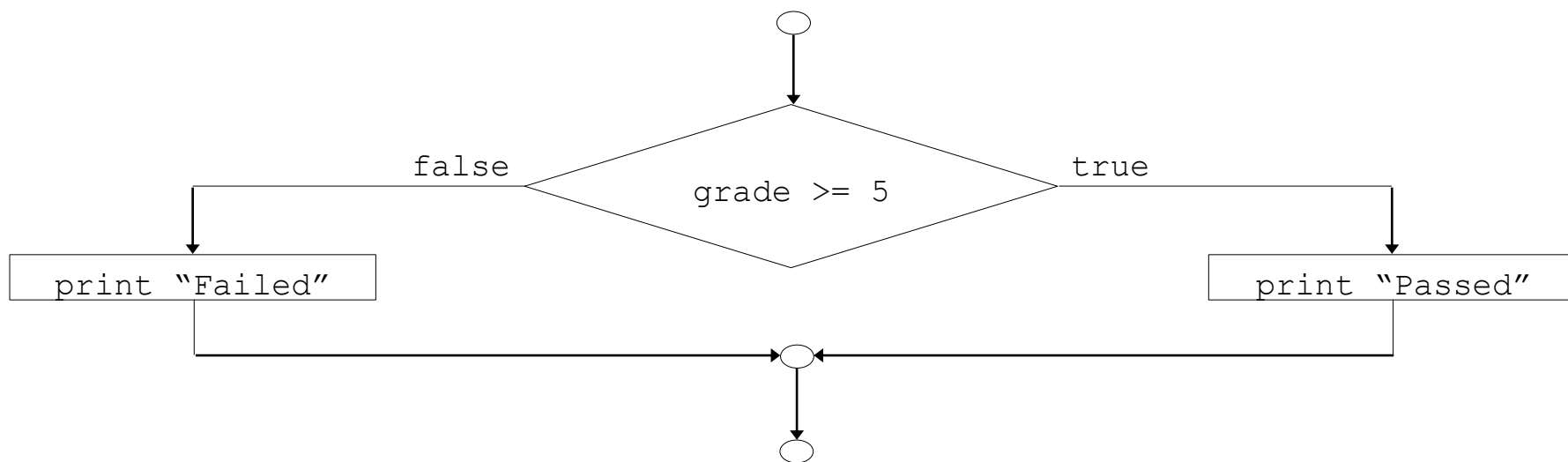
```
grade >= 5 ? printf("Passed\n") : printf("Failed\n" );
```

```
printf( "%s\n", grade >= 5 ? "Passed" : "Failed" );
```



# Η εντολή επιλογής `if...else` (4/10)

- Διάγραμμα ροής της εντολής επιλογής `if...else`





# Η εντολή επιλογής `if...else` (5/10)

- Ένθετες/ «Φωλιασμένες» (nested) εντολές `if...else`
  - Όταν μια εντολή `if` είναι ο προορισμός μιας άλλης εντολής `if` ή `else`, τότε είναι ένθετη μέσα στην πρώτη (εξωτερική) εντολή `if`
  - Παράδειγμα

```
if (count > max) /*εξωτερική if*/  
    if (error) printf("Error, try again."); /*ένθετη if*/
```
  - Στην πράξη σπάνια συναντάται μεγάλο βάθος ένθεσης
  - Ποια `if` σχετίζεται με ποια `else` ;
    - Μια `else` σχετίζεται πάντα με την πλησιέστερη `if` που υπάρχει στο ίδιο τμήμα κώδικα και δεν είναι συσχετισμένη με μια άλλη `if`



# Η εντολή επιλογής `if...else` (6/10)

- Ψευδοκώδικας για ένθετες εντολές `if...else`

*If ο βαθμός του φοιτητή είναι μεγαλύτερος ή ίσος του 9*

*Τύπωσε "A"*

*else*

*If ο βαθμός του φοιτητή είναι μεγαλύτερος ή ίσος του 8*

*Τύπωσε "B"*

*else*

*If ο βαθμός του φοιτητή είναι μεγαλύτερος ή ίσος του 7*

*Τύπωσε "C"*

*else*

*If ο βαθμός του φοιτητή είναι μεγαλύτερος ή ίσος του 6*

*Τύπωσε "D"*

*else*

*Τύπωσε "F"*



# Η εντολή επιλογής `if...else` (7/10)

---

## □ Κώδικας C

```
if (grade >= 9)
    printf("A\n");
else
    if (grade >= 8)
        printf("B\n");
    else
        if (grade >= 7)
            printf("C\n");
        else
            if (grade >= 6)
                printf("D\n");
            else
                printf("F\n");
```



# Η εντολή επιλογής `if...else` (8/10)

---

## □ Κώδικας C

```
if (grade >= 90)
    printf("A\n");
else if (grade >= 80)
    printf("B\n");
else if (grade >= 70)
    printf("C\n");
else if (grade >= 60)
    printf("D\n");
else
    printf("F\n");
```



# Η εντολή επιλογής `if...else` (9/10)

---

## □ Γενική μορφή ένθετων εντολών `if...else`

`if` (έκφραση)

εντολή;

`else if` (έκφραση)

εντολή;

`else if` (έκφραση)

εντολή;

·

·

·

`else`

εντολή;

# Η εντολή επιλογής if...else (10/10)



## □ Παραδείγματα

```
if (p) /*εξωτερική if*/  
    if (q) printf("a and b are true"); /*ένθετη if*/  
    else printf("To which statement does this else  
apply?");
```

```
if (ch=='s')  
{ /*πρώτη if*/  
    printf("Enter a number: \n");  
    scanf("%d", &y);  
    /*δεύτερη if*/  
    if (y) printf("Its square is %d.",y*y);  
}  
    else printf("Make next selection.\n");
```



# Σημερινή διάλεξη

---

- Σύνοψη 1ης διάλεξης
- Τελεστές
- Συναρτήσεις
- Γενικές και τοπικές μεταβλητές
- Παράδειγμα πρόσθεσης ακεραίων
- Ανάπτυξη προγραμμάτων δομημένου προγραμματισμού
  - Αλγόριθμοι
  - Ψευδοκώδικας
- Δομές ελέγχου
  - Η εντολή If
  - Η εντολή If...Else
  - Η εντολή επανάληψης While



# Η εντολή επανάληψης `while` (1/2)

## □ Δομή επανάληψης

- Ο προγραμματιστής ορίζει την επανάληψη μιας ενέργειας για όσο (`while`) κάποια συνθήκη παραμένει αληθής (`true`)

- Ψευδοκώδικας

*While υπάρχουν πράγματα στη λίστα με τα ψώνια*

*Αγόρασε το επόμενο αντικείμενο και σβήσε το από τη λίστα*

- Βρόχος

- Ομάδα εντολών που εκτελεί επανειλημμένα ένας υπολογιστής μέχρι κάποια για όσο (`while`) μια συνθήκη παραμένει `true` (αληθής)
- Ο βρόχος `while` επαναλαμβάνεται μέχρι η συνθήκη να γίνει `false`



# Η εντολή επανάληψης `while` (2/2)

## □ Παράδειγμα

```
int product = 2;  
while ( product <= 1000 )  
    product = 2 * product;
```

