ΕΠΛ 033: ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ

Μάριος Belk, Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

Email: belk@cs.ucy.ac.cy

ΣΥΝΑΡΤΗΣΕΙΣ

Θέμα Διάλεξης: Συναρτήσεις

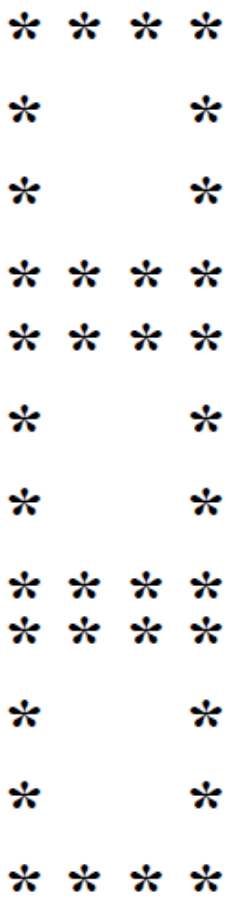
- *Χρησιμότητα Συναρτήσεων*
- *Σύνταξη και Σημασία Συναρτήσεων*
- *Συναρτήσεις Βιβλιοθήκης*

Βήμα προς βήμα

- Κάθε πρόγραμμα στη γλώσσα προγραμματισμού C ξεκινά από την κύρια συνάρτηση που ονομάζεται main. Μια συνάρτηση είναι απλά μια συλλογή εντολών που κάνουν "κάτι".
- Μία συνάρτηση προτού χρησιμοποιηθεί σε κάποιο πρόγραμμα:
  - Ή θα οριστεί μέσα στο ίδιο αρχείο που θα κληθεί
  - Ή, εάν θα οριστεί σε άλλο αρχείο βιβλιοθήκης, θα πρέπει το αρχείο αυτό να εισαχθεί με την εντολή #include στο αρχείο μέσα στο οποίο θα κληθεί.

Παράδειγμα

- Γράψτε ένα πρόγραμμα που να τυπώνει το πιο κατω σχημα:



Παράδειγμα – Λύση 1

```
int main(){
printf("****\n");
printf("*   *\n");
printf("*   *\n");
printf("****\n");

printf("****\n");
printf("*   *\n");
printf("*   *\n");
printf("****\n");

printf("****\n");
printf("*   *\n");
printf("*   *\n");
printf("****\n");
return 0;
}
```

Ροή Ελέγχου

Εκτέλεση ίδιων εντολών σημαίνει ξαναγράψιμο ίδιων εντολών!

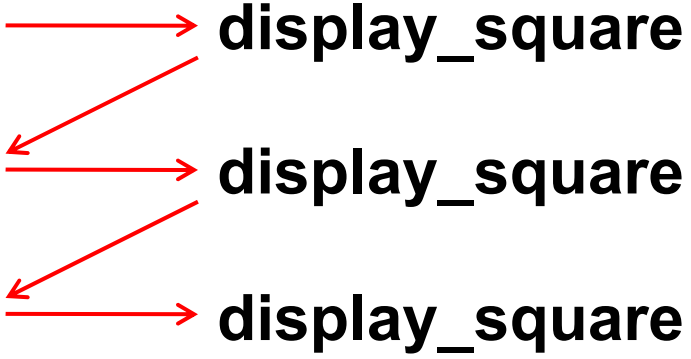
Παράδειγμα – Λύση 2

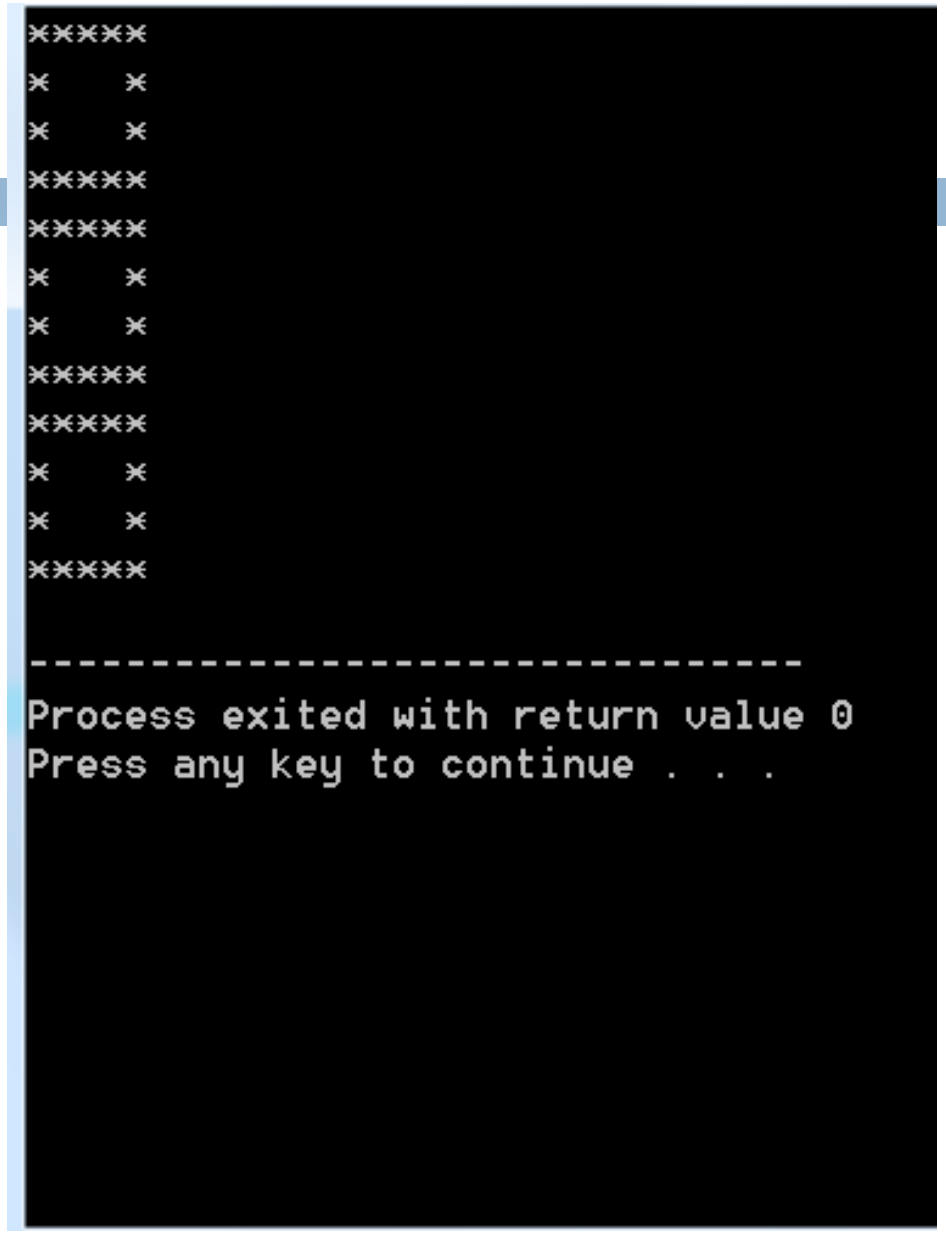
```
void display_square(){
printf(“****\n”);
printf(“*   *\n”);
printf(“*   *\n”);
printf(“****\n”);
 }

int main(){
display_square();

display_square();

display_square();

return 0;
}
```

Ίδιο πρόγραμμα με συνάρτηση

Επαναχρησιμοποίηση ίδιου κώδικα

Ροή Ελέγχου




*****
*   *
*   *
*****
*****
*   *
*   *
*****
*****
*   *
*   *
*****

Process exited with return value 0
Press any key to continue . . .

Συναρτήσεις (Functions)

- Τι είναι;
  - Ένα “μαύρο κουτί” στο οποίο δίνουμε πληροφορίες και μας επιστρέφει απαντήσεις
  - Στην ουσία είναι τμήμα κώδικα που εκτελεί συγκεκριμένη διεργασία
- Γιατί μας ενδιαφέρουν;
  - Αφαιρετικότητα (abstraction): διαχωρισμός ανάμεσα στο τι και το πώς
  - Αρθρωτός σχεδιασμός (modular design)
    - Διευκολύνει ανάπτυξη, κατανόηση και τροποποίηση κυρίως μεγάλων προγραμμάτων
  - Επαναχρησιμοποίηση (reuse)

Είδη Συναρτήσεων

- Συναρτήσεις Βιβλιοθήκης
  - Εγγενείς συναρτήσεις – δεν ορίζονται από τον χρήστη, π.χ. printf, scanf, rand
  - ενσωμάτωση με την εντολή #include της κατάλληλης βιβλιοθήκης όπου η συνάρτηση είναι ορισμένη (π.χ. stdio.h, stdlib.h)
- Συναρτήσεις που ορίζονται από τον προγραμματιστή (user defined functions)

```c
#include <stdio.h>

void display_square()
{
	printf("\n");
	printf("*****\n");
	printf("*****\n");
	printf("*     *\n");
	printf("*     *\n");
	printf("*****\n");
	printf("*****\n");
	printf("\n");
	printf("\n");
}

int main()
{
	display_square();
	display_square();
	display_square();
	return 0;
}
```

```
*****
*****
*     *
*     *
*****
*****


*****
*****
*     *
*     *
*****
*****


*****
*****
*     *
*     *
*****
*****


--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Παραδείγματα Εγγενών Συναρτήσεων

Συνάρτηση	Βιβλιοθήκη	Χρήση	Τ. Εισόδου	Τ. Εξόδου
abs(x)	stdlib.h	Απόλυτη τιμή του x	int	int
fabs(x)	math.h	Απόλυτη τιμή του x	double	double
ceil(x)	math.h	Μικρότερος ακέραιος που είναι μεγαλύτερος του x	double	double
floor(x)	math.h	Μεγαλύτερος ακέραιος που είναι μικρότερος του x	double	double
sin(x)	math.h	Ημίτονο της γωνιάς x	double	double
cos(x)	math.h	Συνημίτονο της γωνιάς x	double	double
tan(x)	math.h	Εφαπτόμενη της γωνιάς x	double	double
exp(x)	math.h	e^x, όπου e = 2.71828	double	double
log(x)	math.h	Φυσικός λογάριθμος του x (ln(x))	double	double
log10(x)	math.h	Λογάριθμος βάσεως 10 του x ($\log_{10}(x)$)	double	double
pow(x,y)	math.h	x υψωμένο στη δύναμη του y (x^y)	double, double	double

Σύνταξη Συναρτήσεων

- Κάθε πρόγραμμα στη C αποτελείται από μια ή περισσότερες συναρτήσεις
  - Κάθε πρόγραμμα έχει μια συνάρτηση main
- Κάθε συνάρτηση πρέπει να ορίζεται
- Μια συνάρτηση πριν αναφερθεί πρέπει να έχει ήδη ***ορισθεί*** (defined) ή ***δηλωθεί*** (declared)
- Δεν μπορούμε να ορίσουμε μια συνάρτηση μέσα σε άλλη συνάρτηση

Σύνταξη Ορισμού Συνάρτησης

Διεπαφή

<τύπος επιστροφής> <όνομα συνάρτησης> (<λίστα παραμέτρων>)

{ αρχή της συνάρτησης

[δηλώσεις μεταβλητών]

[εντολές]

[return <έκφραση>;]

Σώμα συνάρτησης

} τέλος της συνάρτησης

Σύνταξη (συνέχεια)

- <Τύπος τιμής εξόδου>:

  int, char, float, double, void, *σύνθετος*…

- <Όνομα Συνάρτησης>:

  συντακτικά ορθό όνομα

- <Λίστα Παραμετρων>:
  - άδεια ή
  - τύπος μεταβλητή, τύπος μεταβλητή, …

Σύνταξη Ορισμού Συνάρτησης – Παράδειγμα

Διεπαφή

```
int addition  (int a, int b) {
```

```
int result;

result = a + b;

return result;
```

Σώμα συνάρτησης

```
} τέλος της συνάρτησης
```

Κλήση συνάρτησης

```
....
int  main() {
      int a = 10;
      int b = 25;

int c = addition(a, b);
...
return (0);
} τέλος της main
```

Συνάρτηση main χωρίς παραμέτρους

```
void main(){

    printf("Hello out there!\n");

}
```

Συνάρτηση χωρίς παραμέτρους

```
#include <stdio.h>

void display_six_stars()
{
    printf("******\n");
    return;
}

int main()
{
display_six_stars();
display_six_stars();
return 0;
}
```

Συνάρτηση χωρίς παραμέτρους (συνέχεια)

```
void display_six_stars()
{
    printf("******\n");
}
int main(){
    display_six_stars();
    display_six_stars();
    return 0;
}
```

επιστροφή

Κλήση

```
******
******

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Συνάρτηση με 1 Παράμετρο χωρίς τιμή εξόδου (1/2)

```
void display_area (float area)
{
        printf("The area is %f square meters\n", area);
}
```

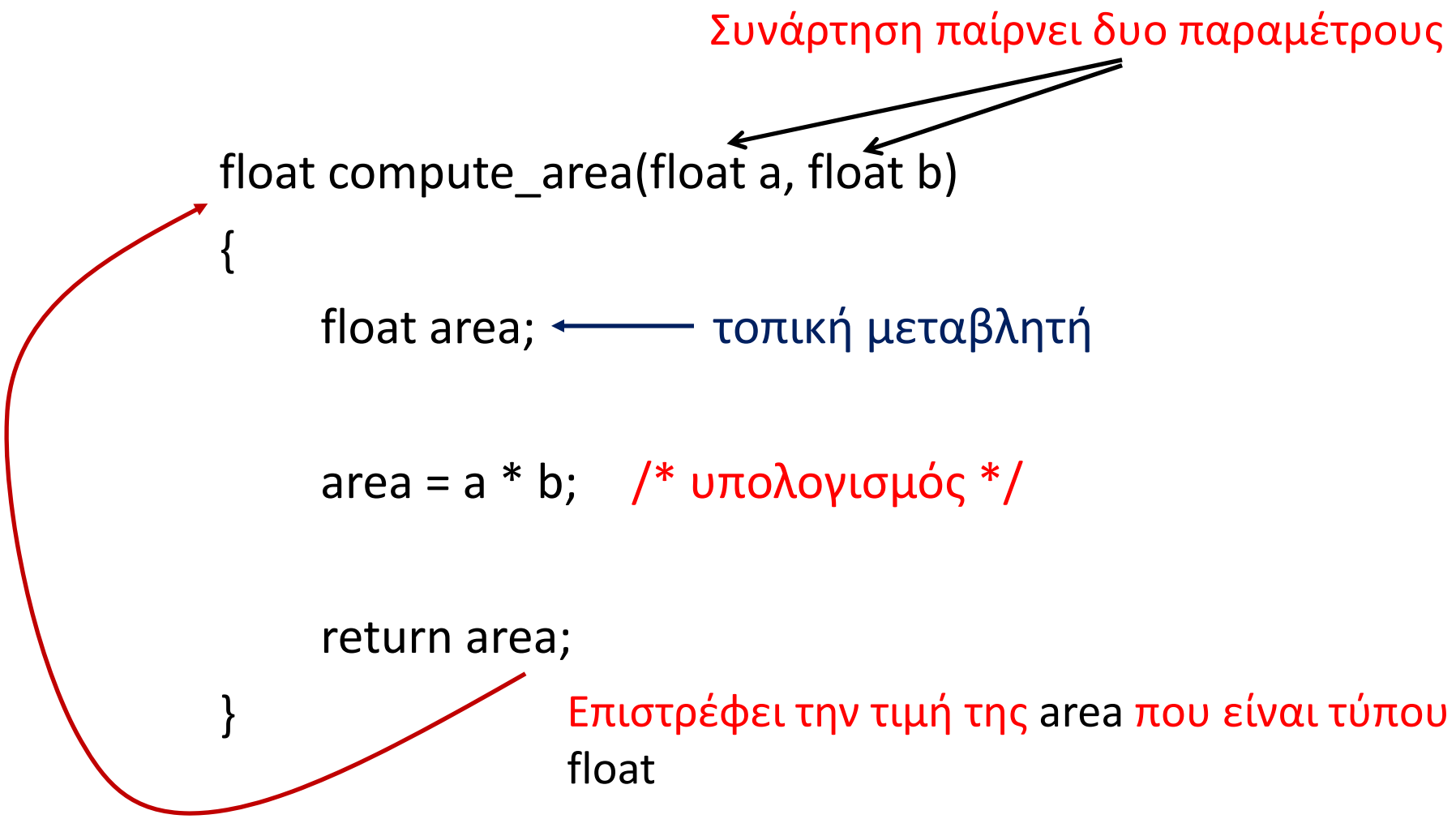
Συνάρτηση με 1 Παράμετρο χωρίς τιμή εξόδου (2/2)

```
 void compute_and_display_mod3(int number)
{
        printf("%dmod3 is %d\n",number, number%3);
}

void compute_and_display_mod3(int number)
{
        int mod3;                   /* τοπική μεταβλητή */
        mod3 = number%3;    /* υπολογισμός */

        printf("%dmod3 is %d\n", number, mod3);
}
```

Συνάρτηση με πολλαπλές παραμέτρους χωρίς τιμή εξόδου

```
void display_area(float length, float width, float area)
{

printf("The area of a rectangle with length %f m and width %f m
is %f sq. m\n", length, width, area);

}
```

Συναρτήσεις με παραμέτρους και τιμή εξόδου



Εναλλακτικά

```
float compute_area(float a, float b)
{
        return a * b;
}
```

Επιστρέφει την τιμή της έκφρασης a*b
που είναι τύπου float

Παραμέτροι

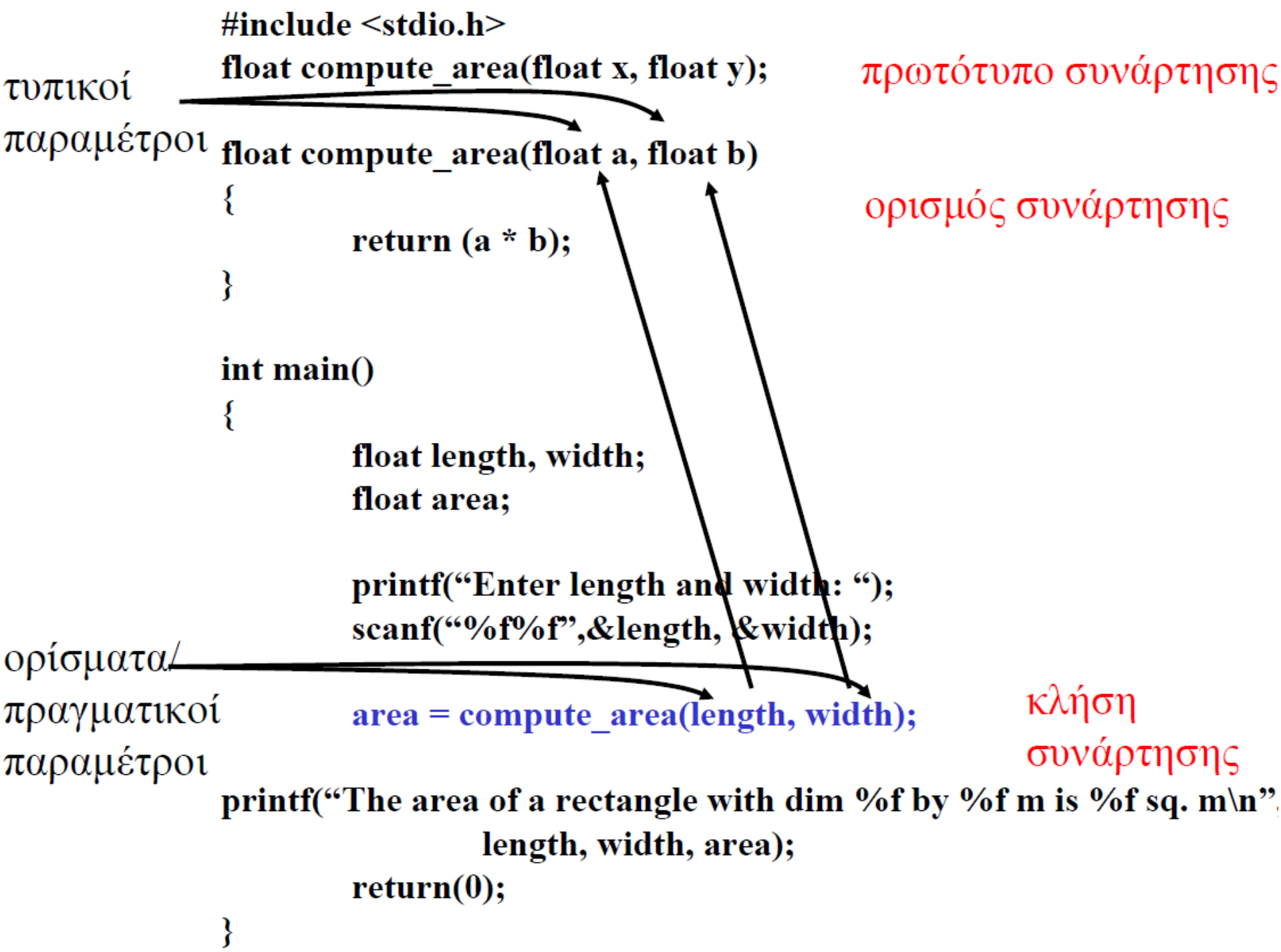
- Επιτρέπουν την επικοινωνιά (διασύνδεση) μεταξύ συναρτήσεων (είσοδο και έξοδο δεδομένων)
- Λίστα παραμέτρων ορίζει τον αριθμό, σειρά και τύπο τιμών που δέχεται μια συνάρτηση
  - Για είσοδο: πέρασμα δια τιμής (τιμή)
  - Για είσοδο/έξοδο: πέρασμα δια αναφοράς (διεύθυνση)- αργότερα

Σημασία Κλήσης

- Δέσμευση μνήμης για παραμέτρους και τοπικές μεταβλητές της συνάρτησης (εάν υπάρχουν)
- Αντιγραφή των τιμών των ορισμάτων (πραγματικοί παραμέτροι) στις τυπικές παραμέτρους (εάν υπάρχουν παραμέτροι)
  - αριθμός, σειρά και τύποι ορισμάτων να ταιριάζουν με τον αριθμό, σείρα και τύπους τυπικών παραμέτρων
- Ξεκινά εκτέλεση από την πρώτη εντολή της συνάρτησης

Σημασία Επιστροφής

- Αποτίμηση της έκφρασης που ακολουθεί το return και αντιγραφή της τιμής εξόδου στο σημείο κλήσης (εάν επιστρέφεται τιμή).
- Συνέχιση εκτέλεσης με την εντολή που ακολουθεί την κλήση
- Αποδέσμευση μνήμης που δεσμεύθηκε όταν έγινε η κλήση. Οι αντίστοιχες μεταβλητές δεν υφίστανται πλέον


#include <stdio.h>
float compute_area(float x, float y);
τυπικοί παραμέτροι
πρωτότυπο συνάρτησης
float compute_area(float a, float b)
{
return (a * b);
}
ορισμός συνάρτησης
int main()
{
float length, width;
float area;
printf("Enter length and width: ");
scanf("%f%f",&length, &width);
ορίσματα/ πραγματικοί παραμέτροι
area = compute_area(length, width);
κλήση συνάρτησης
printf("The area of a rectangle with dim %f by %f m is %f sq. m\n",
length, width, area);
return(0);
}

Δήλωση Συνάρτησης (Πρωτότυπο)

- Σύνταξη Πρωτότυπου Συνάρτησης:

  <τύπος επιστροφής> <όνομα συνάρτησης> (<λίστα παραμέτρων>);

- Σημασία: επιτρέπει χρήση μιας συνάρτησης *πριν* ακόμα ορισθεί

- Καλός προγραμματισμός: λίστα με πρωτότυπα όλων των συναρτήσεων σε ένα πρόγραμμα

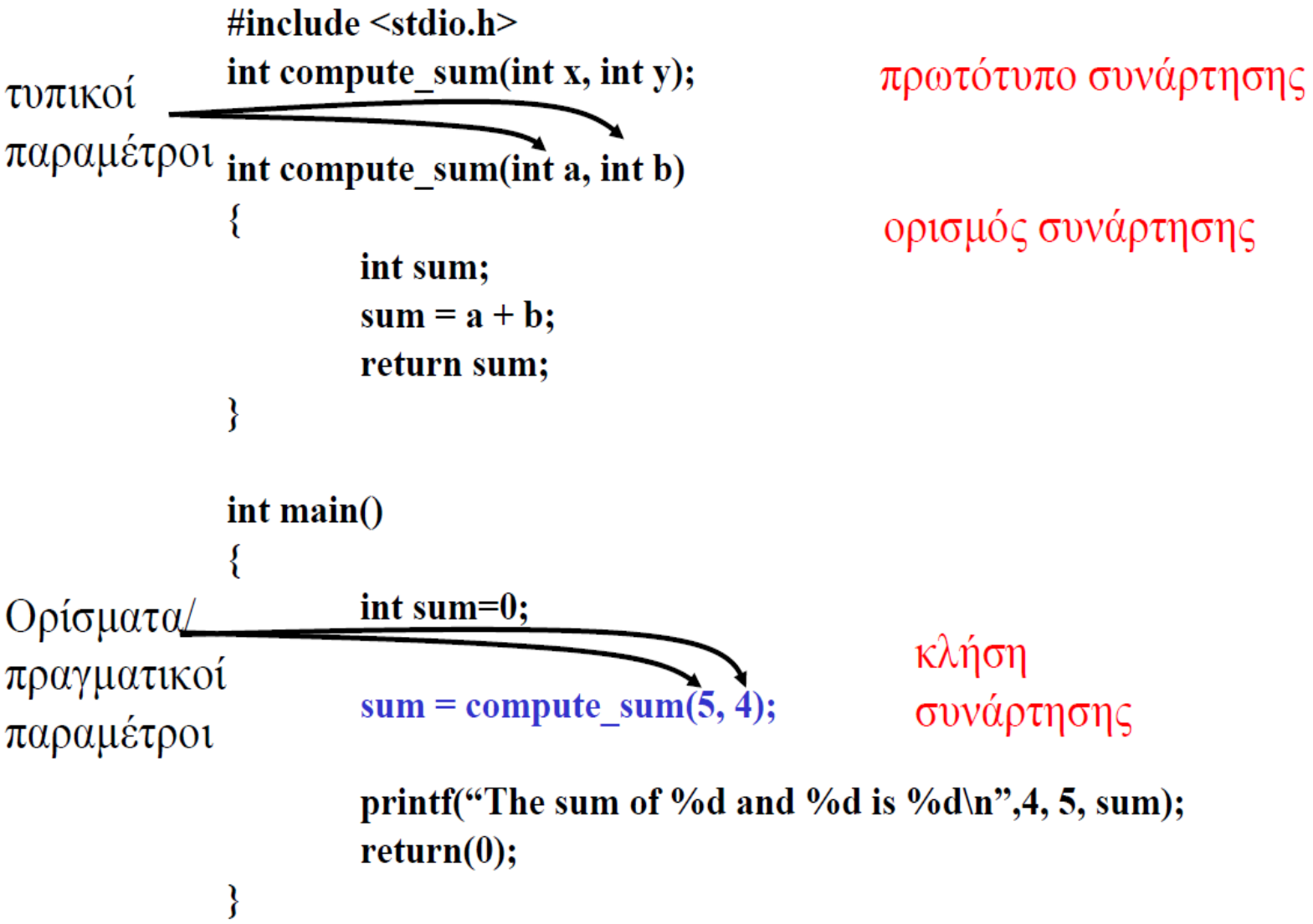
Πρωτότυπα Συναρτήσεων

```
/* σχολια */

/* #include */

/* πρωτότυπα συναρτήσεων */
float compute_area(float length, float width);
void error_message_for_negative_input();
int get_grade(int student_id);

/* σταθερες*/

/* ορισμός συναρτήσεων - μια είναι η main*/
```

Δήλωση συναρτήσεων (Πρωτότυπων)

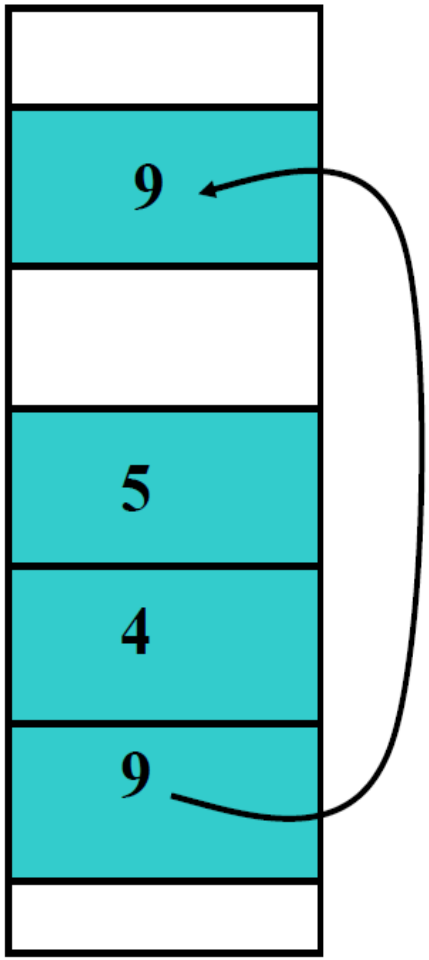
Υλοποίηση συναρτήσεων

Παράδειγμα


#include <stdio.h>
int compute_sum(int x, int y);
τυπικοί παραμέτροι
πρωτότυπο συνάρτησης
int compute_sum(int a, int b)
{
int sum;
sum = a + b;
return sum;
}
ορισμός συνάρτησης
int main()
{
int sum=0;
Ορίσματα/ πραγματικοί παραμέτροι
sum = compute_sum(5, 4);
κλήση συνάρτησης
printf("The sum of %d and %d is %d\n",4, 5, sum);
return(0);
}

Παράδειγμα – (συν.)

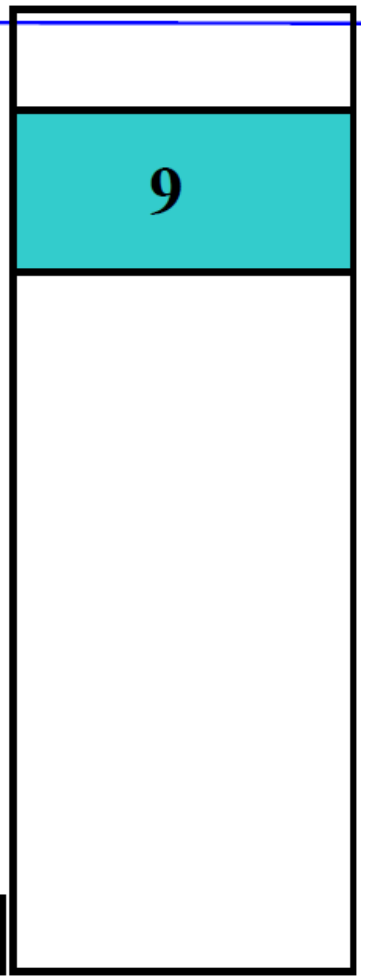
```
#include <stdio.h>
int compute_sum(int x, int y);

int compute_sum(int a, int b)
{
        int sum;
        sum = a + b;
        return sum;
}

int main()
{
        int sum=0;

        sum = compute_sum(5, 4);

        printf("The sum of %d and %d is %d\n",4, 5, sum);
        return(0);
}
```



Παράδειγμα – (συν.)

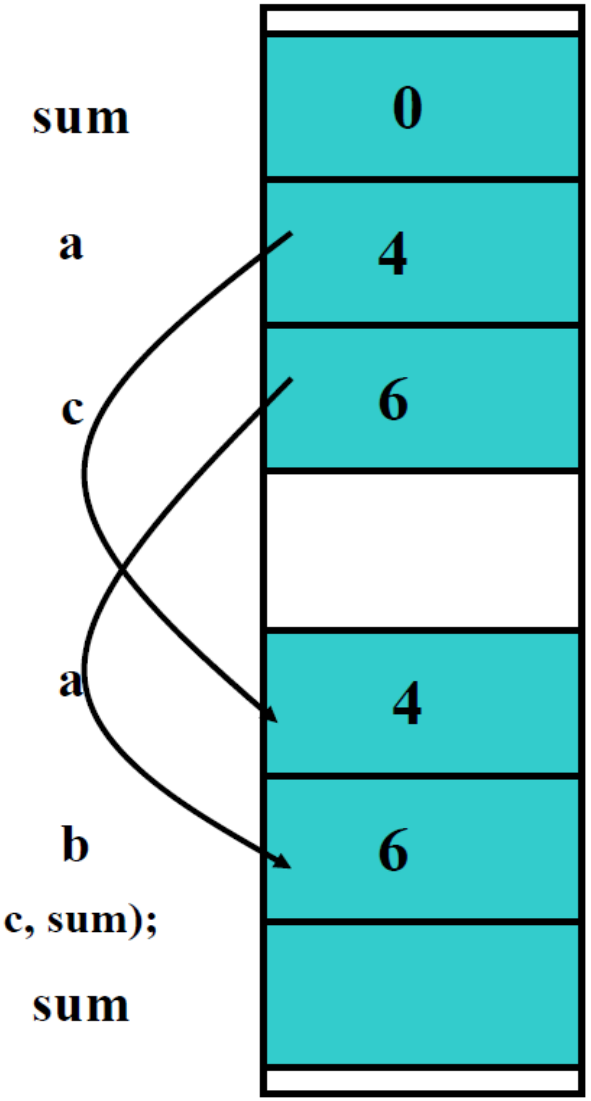
```
#include <stdio.h>
int compute_sum(int x, int y);

int compute_sum(int a, int b)
{
        int sum;
        sum = a + b;
        return sum;
}

int main()
{
        int sum=0;

        sum = compute_sum(5, 4);

        printf("The sum of %d and %d is %d\n",4, 5, sum);
        return(0);
}
```

sum



Παράδειγμα – (συν.)

```
#include <stdio.h>
int compute_sum(int x, int y);

int compute_sum(int a, int b)
{
        int sum;
        sum = a + b;
        return sum;
}

int main()
{
        int sum=0;
        int a=4, c=6;
        sum = compute_sum(a, c);

        printf("The sum of %d and %d is %d\n",a, c, sum);
        return(0);
}
```



Εμβέλεια Μεταβλητής (scope)

- Το τμήμα του προγράμματος που μπορεί μια μεταβλητή να χρησιμοποιηθεί
  - local (τοπικές):
    - Δηλώνονται στην αρχή ενός “programming block” {..}
    - Χρήση οπουδήποτε μετά τον ορισμό μέσα στο block
  - global (σφαιρικές/καθολικές): ΑΠΑΓΟΡΕΥΟΝΤΑΙ ΣΤΟ ΕΠΛ033
    - Δηλώνονται έξω από συναρτήσεις
    - Χρήση οπουδήποτε μετά τον ορισμό

Παράδειγμα

- Γράψετε μια συνάρτηση με την ακόλουθη διεπαφή:

  void display_time (int time);

- Η διαδικασία παίρνει μια παράμετρο, την ώρα (time), σε 24ωρη μορφή και τυπώνει την ώρα και τα λεπτά χωρισμένα με **:** π.χ. εάν η τιμή της παραμέτρου εισόδου είναι 1256, η έξοδος θα είναι

  The time is 12:56

Παράδειγμα – Λύση

```
void display_time(int time)
{
        /* compute hours */
        /* compute minutes */
        /* display hours:minutes*/
}
void display_time(int time)
{
            int hours,minutes;
            /* compute hours */
            hours = time / 100;
            /* compute minutes */
            minutes = time % 100;
            /* display hours:minutes*/
            printf("The time is %02d: %02d\n", hours, minutes);
}
```

Περίληψη

- Χρησιμότητα Συναρτήσεων
  - Αφαιρετικότητα
  - Άρθρωση
  - Επαναχρησιμοποίηση
- Συναρτήσεις Βιβλιοθήκης
- (User defined) Συναρτήσεις
  - Σύνταξη και Σημασία
  - Ορισμός, Δήλωση, Κλήση Συνάρτησης
  - Παράμετροι
  - Επιστροφή τιμής
  - Εμβέλεια Μεταβλητών