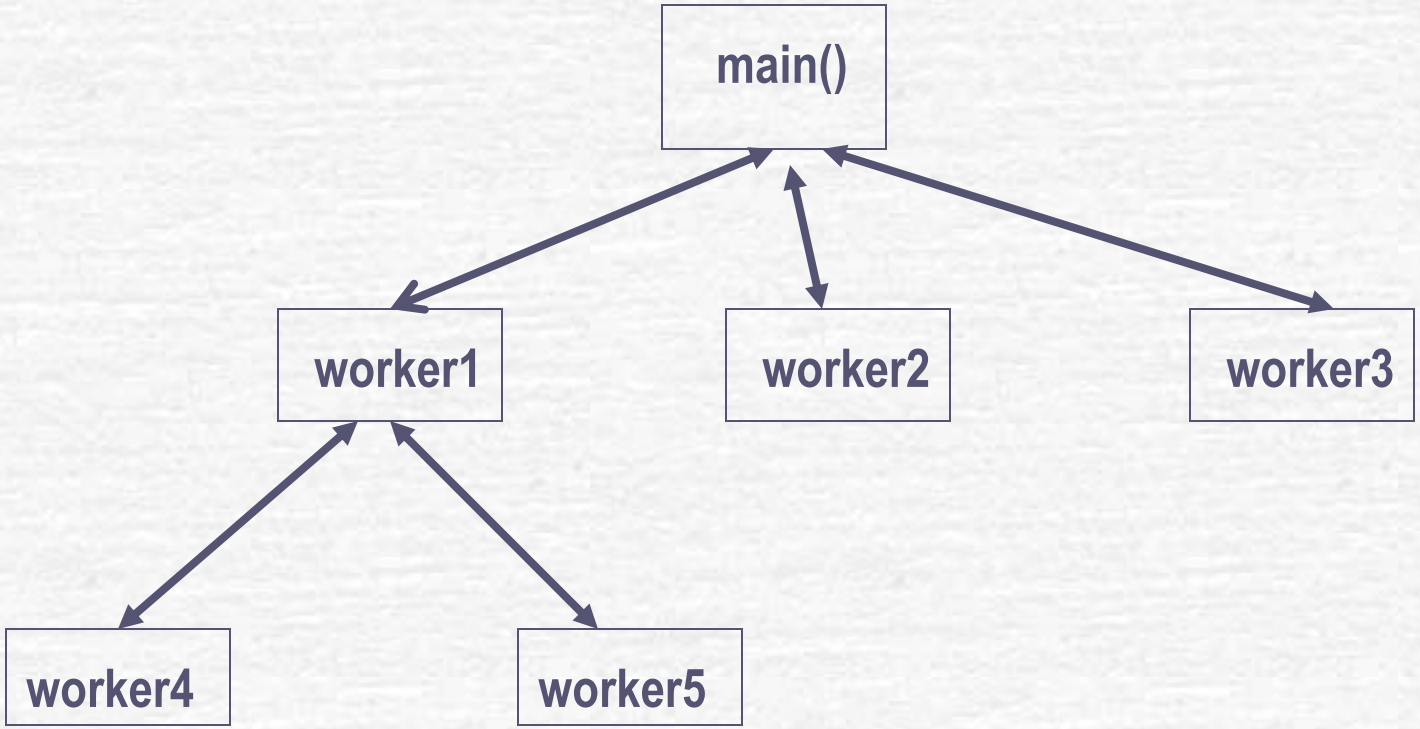
ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ – ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Συναρτήσεις (Functions)



Υποπρογράμματα

- Αποτελούν τον τρόπο εφαρμογής (υλοποίησης) της τμηματοποίησης σε ένα πρόγραμμα.
- Κάθε υποπρόγραμμα είναι ένα αυτόνομο τμήμα (μικρό πρόγραμμα)
- Ο συνδυασμός των υποπρογραμμάτων συνιστά το (αρχικό) πρόγραμμα.
- Πλεονεκτήματα
  - Αποφυγή επαναλήψεων
  - Αύξηση επαναχρησιμοποίησης
  - Βελτίωση αναγνωσιμότητας
  - Ευκολότερη συντήρηση

Συναρτήσεις στη C

- Πρόγραμμα C = σύνολο συναρτήσεων
- Η συνάρτηση main αντιπροσωπεύει το κυρίως πρόγραμμα, δηλ. τον τρόπο με τον οποίο συνδυάζονται οι υπόλοιπες συναρτήσεις για τη λύση του προβλήματος.
- Μέρη συνάρτησης
  - Κεφαλίδα = η διεπαφή της συνάρτησης (όνομα, είσοδος, έξοδος)
  - Σώμα = υλοποίηση/ορισμός της συνάρτησης
- Προτάσεις συνάρτησης
  - Δήλωση – Κλήση - Ορισμός

Δήλωση Συνάρτησης

- Προσδιορίζεται η διεπαφή, ο τρόπος αναφοράς στη συνάρτηση:

  <τύπος> <όνομα – συν> ([<παράμετροι>]); Όπου

  <παράμετροι>:=<τύπος1> [<όνομ-παρ1>], …, <τύποςN> [<όνομ-παρN]

  τυπικά ορίσματα (ή είσοδοι)

  τυπικά αποτελέσματος (εξόδου)

- Αν η συνάρτηση δεν επιστρέφει κάποια τιμή, τότε χρησιμοποιείται σαν τύπος αποτελέσματος η λέξη κλειδί void.

Παραδείγματα

```
int max (int a, int b);

int min (int, int);

double exp ( double m, int n);

void swap (int a, int b);

char *getptr (char * str, char ch);

char *getptr (char str [ ], char ch);
```

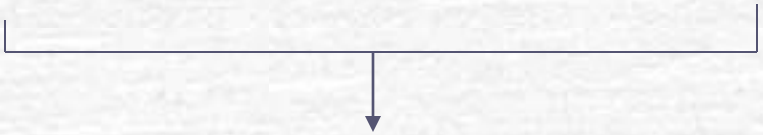
Παράδειγμα

```
#include <stdio.h>
int square(int y); /* αρχέτυπο συνάρτησης */
int main ( )
{
    int x;
    for (x=1; x<=10; x++)
            printf("%d", square(x));
    printf("\n");
    return 0;
}
/* ορισμός συνάρτησης */
int square( int y)
{
    return y * y;
}
```

Κλήση Συνάρτησης

- Καλείται η συνάρτηση για εκτέλεση με συγκεκριμένα ορίσματα

  <όνομα – συν> (<ορισ1>, <ορισ2>, …, <ορισN>);

  πραγματικά ορίσματα (σταθερές, μεταβλητές, εκφράσεις|)

- Τα πραγματικά ορίσματα πρέπει να είναι του ίδιου αριθμού και τύπου με τα τυπικά ορίσματα

Παραδείγματα

```
swap ( x, y);

draw_circle (a/2.0, 2.0 * b, c);

max_num = max(num1, num2);

min_num = (num, 5);

x = y + max (num1/2.0, 3.0*num2);

printf ( “ο μέγιστος είναι: %d\n”, max(x1,x2));
```

Ορισμός Συνάρτησης

- Για κάθε μη ενσωματωμένη (δική μας) συνάρτηση πρέπει να ορίσουμε το σώμα της στο πρόγραμμα (μετά την main)

```
<τύπος> (<όνομα - συν> (<παράμετροι>)
        {
                <δηλώσεις τοπικών μεταβλητών>
                <προτάσεις>
        }
```

Παράδειγμα (1)

```
double embadon (double platos, double mikos)
{
double apotelesma;
apotelesma = platos * mikos;
platos=15;
 return (apotelesma);
}
```

τυπικές παράμετροι

τοπική μεταβλητή

Επιστροφή ελέγχου και τιμής στην καλούσα συνάρτηση

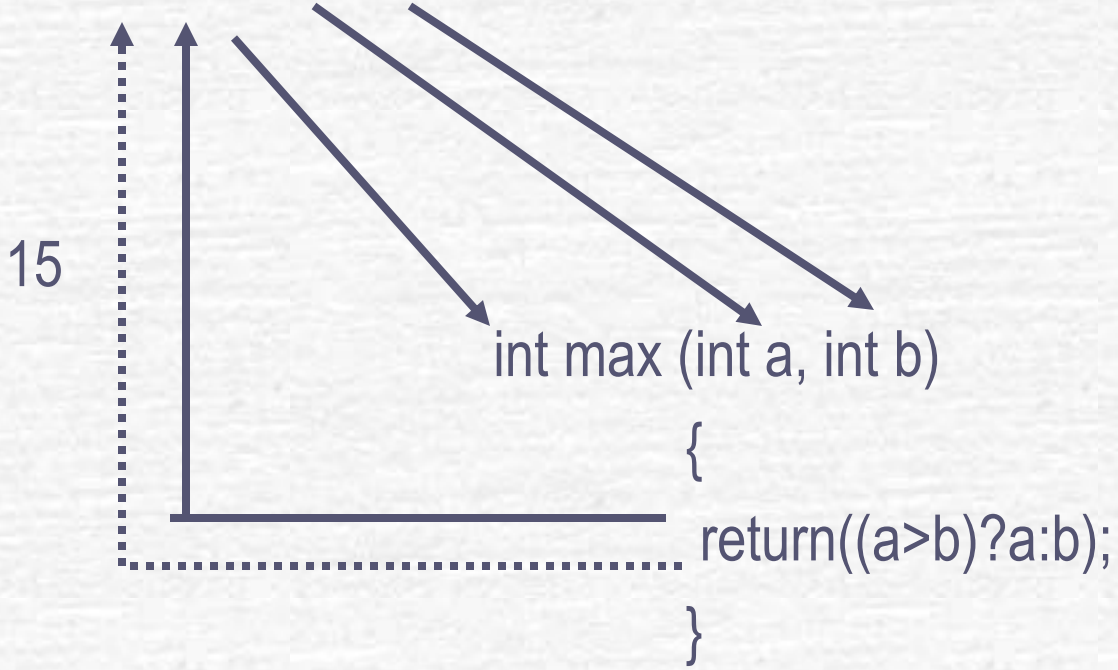
Παράδειγμα (2)

```
int max (int a, int b )
{
    int max;
    max = ( a > b) ? a : b;
    return (max);
}

int max( int a, int b)
{
    return ((a > b) ? a : b);
}
```

Μηχανισμός Κλήσης (1)

max_num = max (max (15, 13), 17);



Αναδρομή (Recursion) (1)

- Ορισμός συνάρτησης μέσω κλήσης του εαυτού της
- Μία τεχνική επίλυσης προβλημάτων
- Π.χ. εύρεση αθροίσματος 1+2+…+n

- Κλασσική λύση

```
int sum (int n) {
s=0;
for (i=1; i<=n; i++)
  s=s+i;
return s;}
```

Αναδρομή (Recursion) (2)

- Αναδρομική λύση

  sum(n) = sum (n-1) + n

```
int sum (int n) {
if  (n<=1)
   return n;
else
   return (sum(n-1)+n);}
```