



ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Εμβέλεια Μεταβλητών

- Εμβέλεια = το τμήμα του προγράμματος στο οποίο έχει ισχύ ή είναι ορατή η μεταβλητή.
- Πότε μας αφορά η εμβέλεια;
Όταν έχουμε μεταβλητές με το ίδιο όνομα.

Τύποι Εμβέλειας (1)

- Εξωτερικές μεταβλητές

■ Γενικές ή καθολικές μεταβλητές (global variables)

- Δηλώνονται έξω και πάνω από κάθε συνάρτηση (και την main), στο τμήμα δηλώσεων μεταβλητών.
- Η εμβέλειά τους είναι όλο το αρχείο του πηγαίου κώδικα που ανήκει η δήλωση, από το σημείο της δήλωσης και κάτω (εμβέλεια αρχείου)

Τύποι Εμβέλειας (2)

- Εσωτερικές ή αυτόματες μεταβλητές
 - Τοπικές μεταβλητές (local variables)
 - Δηλώνονται μέσα σε μία συνάρτηση.
 - Η εμβέλειά τους είναι το σώμα της συνάρτησης (εμβέλεια μπλοκ)
 - Τυπικές παράμετροι συναρτήσεων
 - Η εμβέλειά τους είναι το σώμα της συνάρτησης (εμβέλεια μπλοκ)
 - Μεταβλητές σύνθετης πρότασης
 - Δηλώνονται μέσα σε μία σύνθετη πρόταση
 - Η εμβέλειά τους είναι το σημείο της δήλωσης μέχρι το τέλος της σύνθετης πρότασης (εμβέλεια μπλοκ).
 - Στατικές Μεταβλητές
 - Δηλώνονται σε κάποια (ες) συνάρτηση (εις) (μετά την main) με τη λέξη κλειδί static πριν από τον τύπο μεταβλητής.
 - Η εμβέλειά τους είναι το αρχείο του πηγαίου κώδικα που ανήκει η δήλωση, από το σημείο της δήλωσης και κάτω (εμβέλεια αρχείου)

Κανόνες Εμβέλειας Μεταβλητών

- Μεταβλητές με το ίδιο όνομα επιτρέπονται μόνο όταν έχουν διαφορετική εμβέλεια
- Μεταβλητή με μικρότερη εμβέλεια αποκρύπτει πιθανώς ομώνυμες μεταβλητές μεγαλύτερης εμβέλειας.

Εμβέλεια Συνάρτησης

- Οι συναρτήσεις, όπως και οι μεταβλητές έχουν εμβέλεια
- Η εμβέλεια μίας συνάρτησης εκτείνεται από το σημείο της δήλωσής της μέχρι το τέλος του προγράμματος.
- Αν μία συνάρτηση δηλωθεί static, τότε η εμβέλειά της περιορίζεται το αρχείο που δηλώθηκε.
- Μεταβλητή με μικρότερη εμβέλεια αποκρύπτει πιθανώς ομώνυμες μεταβλητές μεγαλύτερης εμβέλειας.

Διάρκεια Μεταβλητής

- Ο χρόνος δέσμευσης της μνήμης που περιέχει την τιμή της μεταβλητής
- Καθολική μεταβλητή: διάρκεια εκτέλεσης προγράμματος (πλήρης διάρκεια).
- Τοπική μεταβλητή: διάρκεια εκτέλεσης συνάρτησης (περιορισμένη διάρκεια)
- Τυπική παράμετρος: διάρκεια εκτέλεσης συνάρτησης (περιορισμένη διάρκεια)
- Στατική τοπική μεταβλητή: διάρκεια εκτέλεσης προγράμματος (πλήρης διάρκεια)

Αρχικοποίηση Μεταβλητών

- Μία τοπική μεταβλητή περιορισμένης διάρκειας αρχικοποιείται με κάθε είσοδο στο μπλοκ (συνάρτηση) που ορίζεται
- Μία τοπική μεταβλητή πλήρους διάρκειας αρχικοποιείται με την έναρξη εκτέλεσης του προγράμματος

- Π.χ. `static int num;`

```
func (int) {
```

```
    static int count = 0;
```

```
    int num = 100; ... }
```


Παράδειγμα

```
#include <stdio.h>  
void increment(void)
```

```
main() {  
int j=0;  
increment ( );  
increment ( );  
increment ( );  
printf(“%d”, j}
```

```
void increment(void) {  
int j=2;  
static int k=2;  
printf(“j: %d\t k: %d\n”, j++, k++); }
```

Πέρασμα Παραμέτρων

- Κατ' αξία ή τιμή (by value)

Η συνάρτηση δουλεύει σε αντίγραφα των πραγματικών παραμέτρων

- Κατ' αναφορά (by reference)

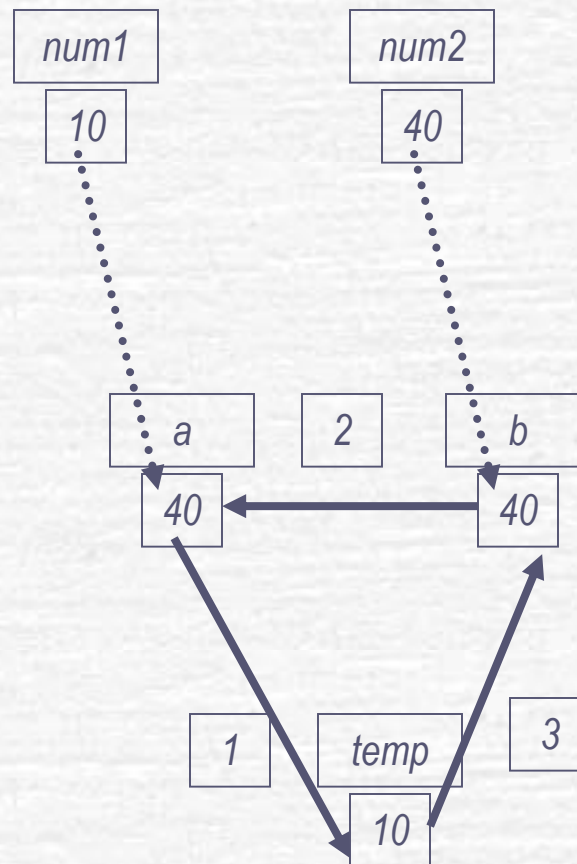
Η συνάρτηση δουλεύει στις πραγματικές παραμέτρους (μόνο για πίνακες)

Παράδειγμα (1)

```
void swap (int a, int b)
{ int temp;
  temp=a;
  a=b;
  b=temp;}
```

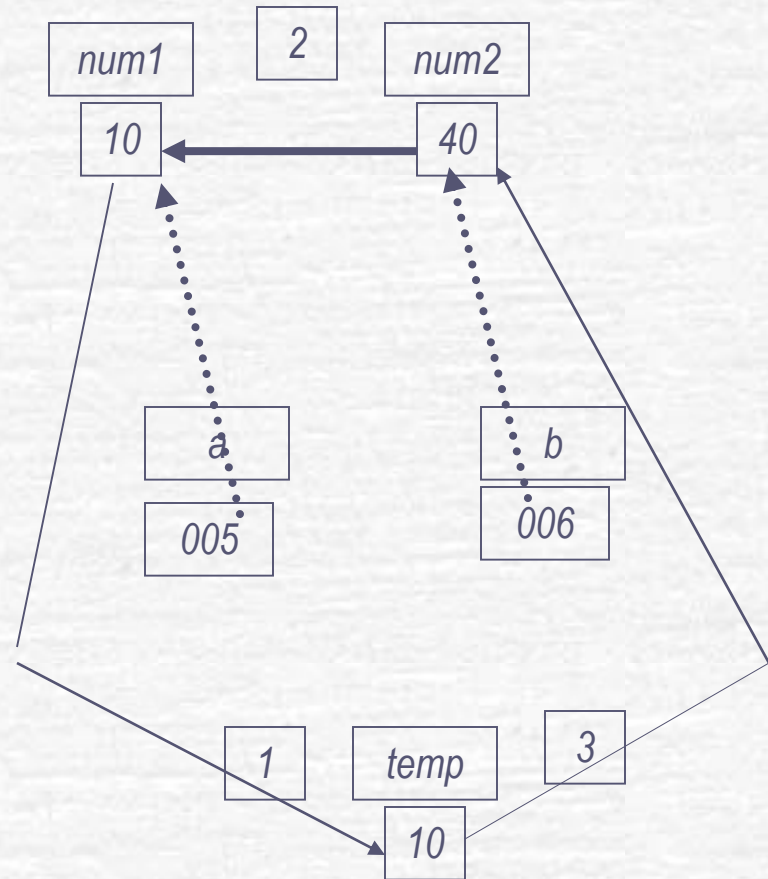
```
swap (num1, num2);
```

Ενώ οι τιμές των a,b αλλάζουν,
οι τιμές των num1, num2
παραμένουν αμετάβλητες



Παράδειγμα (2)

```
void swap (int *a, int * b)  
{ int temp;  
  temp=* a;  
  *a=*b;  
  *b=temp;}  
  
swap (&num1, &num2);
```



Πολλαπλά αρχεία

- Σε πολλαπλά αρχεία υπάρχει ΜΟΝΟ ένα αρχείο με τη `main()`
- Μία εξωτερική μεταβλητή μπορεί να χρησιμοποιηθεί από περισσότερα από ένα αρχεία αν και μόνο εάν η δήλωσή της στα άλλα αρχεία συνοδεύεται από `keyword extern`
- Μία συνάρτηση θα πρέπει να δηλωθεί σε κάθε αρχείο που τη χρησιμοποιεί, ορίζεται όμως ΜΟΝΟ σε ένα
- Αν μία συνάρτηση ή μεταβλητή δηλωθεί `static`, τότε η εμβέλειά της περιορίζεται αναγκαστικά στο αρχείο που δηλώθηκε.

Οδηγίες (<http://stackoverflow.com/questions/1433204/how-do-i-use-extern-to-share-variables-between-source-files-in-c>)

- Ένα header file περιέχει μόνο `extern` δηλώσεις μεταβλητών — ποτέ `static`
- Για κάθε μεταβλητή μόνο ένα header file τη δηλώνει (SPOT — Single Point of Truth).
- Ένα αρχείο πηγαίου κώδικα δεν περιέχει `extern` δηλώσεις μεταβλητών — τα αρχεία πηγαίου κώδικα πάντα περιλαμβάνουν το (μόνο) αρχείο κεφαλίδα που τη δηλώνει.
- For any given variable, exactly one source file defines the variable, preferably initializing it too. (Although there is no need to initialize explicitly to zero, it does no harm and can do some good, because there can be only one initialized definition of a particular global variable in a program).
- The source file that defines the variable also includes the header to ensure that the definition and the declaration are consistent.
- A function should never need to declare a variable using `extern`.
- Avoid global variables whenever possible — use functions instead.