

Δείκτες 1/4

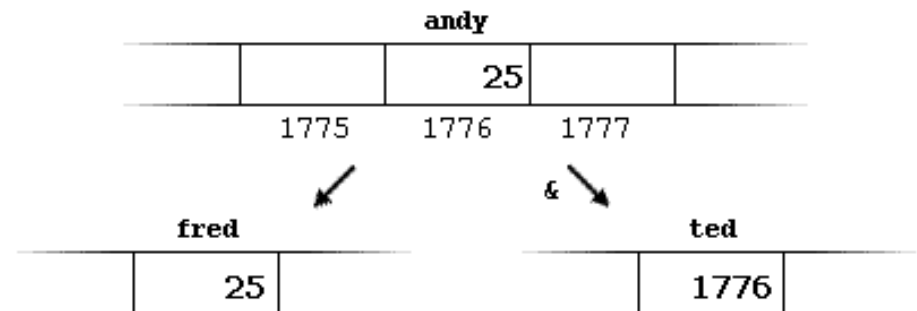
□ Σύμβαση

- Θεωρούμε ότι η μνήμη του υπολογιστή είναι μια διαδοχή από κελιά του 1 byte, κάθε ένα από τα οποία έχει μια μοναδική διεύθυνση

□ Τελεστής &

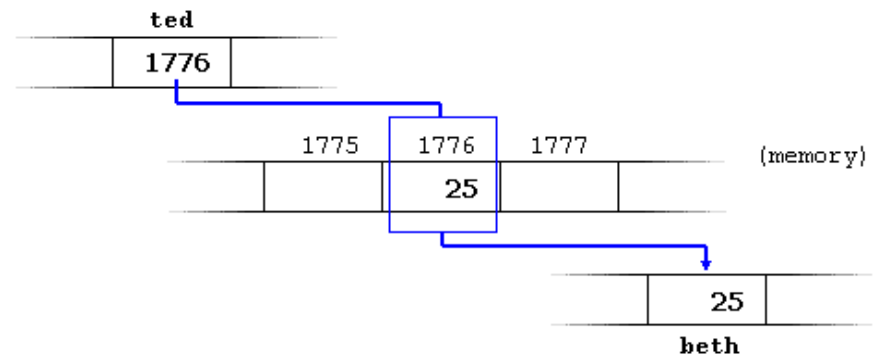
- Το λειτουργικό σύστημα αναλαμβάνει την αποθήκευση των μεταβλητών στη μνήμη. Η δ/νση στην οποία έχει τοποθετηθεί μια μεταβλητή δίνεται με τη βοήθεια του **&** (**διεύθυνση του...**)
- Έστω η μεταβλητή **andy**, που έχει τοποθετηθεί στη δ/νση **1776**

```
■ andy = 25;  
  fred = andy;  
  ted = &andy;
```



Δείκτες 2/4

- Η μεταβλητή που αποθηκεύει τη δ/νση μιας άλλης μεταβλητής καλείται **δείκτης**
- **Τελεστής ***
 - Χρησιμοποιώντας ένα δείκτη μπορούμε να προσπελάσουμε απευθείας την τιμή της μεταβλητής στην οποία δείχνει ο δείκτης, με τη βοήθεια του τελεστή * (**τιμή της δν/σης που δείχνει ο...**)
 - Στο προηγούμενο παράδειγμα:
 - `beth = *ted;`



Δείκτες 3/4

□ Δήλωση δεικτών

- Είναι απαραίτητη η δήλωση του τύπου κάθε δείκτη

- `type * pointer_name;`

- `int * number;`

- `char * character`

- `float * greatnumber;`

- **Προσοχή!** Ο τελεστής `*` στις παραπάνω δηλώσεις σημαίνει ότι οι εν λόγω μεταβλητές είναι δείκτες των αντίστοιχων τύπων, ενώ στην εντολή `beth = *ted` έχει διαφορετική σημασία.

ΔΕΙΚΤΕΣ 4/4

```
#include <iostream.h>
```

```
value1==10 / value2==20
```

```
int main ()
```

```
{
```

```
    int value1 = 5, value2 = 15;
```

```
    int *p1, *p2;
```

```
    p1 = &value1;
```

```
    p2 = &value2;
```

```
    *p1 = 10;
```

```
    *p2 = *p1;
```

```
    p1 = p2;
```

```
    *p1 = 20;
```

```
    cout << "value1==" << value1 << "/ value2==" << value2;
```

```
    return 0;
```

```
}
```

Δείκτες και πίνακες 1/2

- Η έννοια του πίνακα είναι άμεσα συνδεδεμένη με αυτή των δεικτών. Το όνομα ενός πίνακα ισοδυναμεί με τη διεύθυνση του πρώτου του στοιχείου
 - `int numbers [20];`
 - `int * p;`
 - `p = numbers;`
- Διαφορά του **p** και του **numbers**;
 - Η **p** είναι μια μεταβλητή δείκτη, ενώ η **numbers** είναι σταθερά δείκτη. Αυτό σημαίνει ότι δεν είναι νόμιμη η ανάθεση:
`numbers = p;`
- `a[5] = 0;`
`*(a+5) = 0;` Ισοδύναμες εκφράσεις

Δείκτες και πίνακες 2/2

```
#include <iostream.h>
int main ()
{
    int numbers[5];
    int * p;
    p = numbers;
    *p = 10;
    p++;
    *p = 20;
    p = &numbers[2];
    *p = 30;
    p = numbers + 3;
    *p = 40;
    p = numbers;
    *(p+4) = 50;
    for (int n=0; n<5; n++)
        cout << numbers[n] << ", ";
    return 0;
}
```

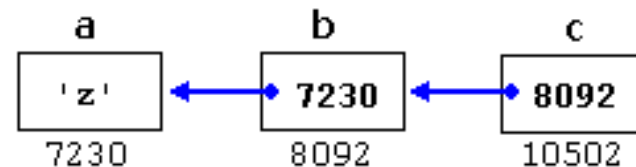
10, 20, 30, 40, 50,

Δείκτες σε δείκτες

- Η C++ επιτρέπει τη χρήση δεικτών που δείχνουν σε δείκτες, οι οποίοι με τη σειρά τους δείχνουν σε δεδομένα. Αυτή η λειτουργία επιτυγχάνεται με τη χρήση ενός αστερίσκου για κάθε επίπεδο αναφοράς

```
char a;  
char * b;  
char ** c;  
a = 'z';  
b = &a;  
c = &b;
```

- Υποθέτουμε ότι η ανάθεση στη μνήμη είναι στις θέσεις **7230, 8092** και **10502**;



Δείκτες void 1/2

- Οι δείκτες **void** είναι ειδικού τύπου δείκτες που μπορεί να δείχνουν σε οποιοδήποτε τύπο δεδομένων, με μοναδικό περιορισμό ότι τα δεδομένα στα οποία δείχνουν δεν μπορούν να αναφερθούν άμεσα, εφόσον το μέγεθός τους είναι ακαθόριστο.
- Για το λόγο αυτό πρέπει να χρησιμοποιούμε μετατροπείς τύπου, ώστε ένας δείκτης **void** να μετατραπεί σε δείκτη συγκεκριμένου τύπου στον οποίο να μπορούμε να αναφερθούμε.
- Χρησιμότητα: πέρασμα γενικών παραμέτρων σε συναρτήσεις.

ΔΕΙΚΤΕΣ void 2/2

6, 10, 13

```
#include <iostream.h>
void increase (void* data, int type)
{
    switch (type)
    {
        case sizeof(char) : (*(char*)data)++; break;
        case sizeof(short): (*(short*)data)++; break;
        case sizeof(long)  : (*(long*)data)++; break;
    }
}
int main ()
{
    char a = 5; short b = 9; long c = 12;
    increase (&a, sizeof(a));
    increase (&b, sizeof(b));
    increase (&c, sizeof(c));
    cout << (int) a << ", " << b << ", " << c;
    return 0;
}
```

Δείκτες σε συναρτήσεις 1/2

- Η C++ επιτρέπει λειτουργίες με δείκτες σε συναρτήσεις. Η ιδιότητα αυτή είναι ιδιαίτερα χρήσιμη για το πέρασμα μιας συνάρτησης ως παραμέτρου σε μια άλλη συνάρτηση.

Δείκτες σε συναρτήσεις 2/2

```
#include <iostream.h>

int addition (int a, int b)
{ return (a+b); }

int subtraction (int a, int b)
{ return (a-b); }

int (*minus) (int,int) = subtraction;

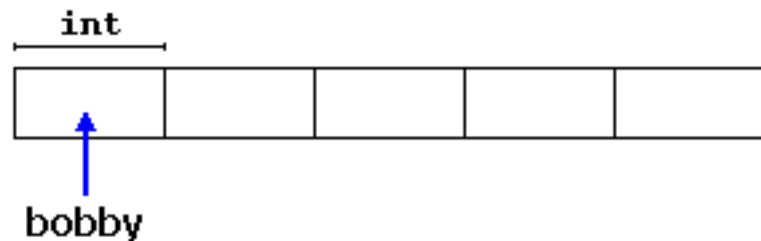
int operation (int x, int y, int
(*functocall) (int,int))
{ int g;
  g = (*functocall) (x,y);
  return (g); }
```

```
int main ()
{
  int m,n;
  m = operation (7, 5, addition);
  n = operation (20, m, minus);
  cout <<n;
  return 0;
}
```

Δυναμική μνήμη 1/3

- Ανάγκη για μεταβλητή ποσότητα μνήμης κατά τη διάρκεια εκτέλεσης του προγράμματος
- Για τη λειτουργία δυναμικής μνήμης η C++ ενσωματώνει τους τελεστές **new** και **delete**
- Με τον τελεστή **new** αιτούμαστε ποσότητα μνήμης
 - **pointer = new type** ή
 - **pointer = new type [elements]**
 - π.χ.

```
int * bobby;
```
 - ```
bobby = new int
```



# Δυναμική μνήμη 2/3

14

- Με τον τελεστή **delete** αποδεσμεύουμε ποσότητα μνήμης που δεν χρησιμοποιούμε πλέον
- **delete pointer** ή
- **delete [] pointer**

```
#include <iostream>

int main()
{
 int n;
 cout << "What array size ? ";
 cin >> n;
 int *x = new int[n];
 for(int k = 0; k < n; k++)
 x[k] = k*k;
 delete[] x;
}
```

# Δυναμική μνήμη 3/3

```
#include <iostream.h>
#include <stdlib.h>

int main ()
{ char input [100]; int i,n; long * l;
 cout << "How many numbers do you want to type in? ";
 cin.getline (input,100); i=atoi (input);
 l= new long[i];
 if (l == NULL) exit (1);
 for (n=0; n<i; n++)
 {
 cout << "Enter number: ";
 cin.getline (input,100);
 l[n]=atol (input);
 }
 cout << "You have entered: ";
 for (n=0; n<i; n++)
 cout << l[n] << ", ";
 delete[] l;
 return 0;}
```

```
How many numbers do you want to
type in? 5
Enter number : 75
Enter number : 436
Enter number : 1067
Enter number : 8
Enter number : 32
You have entered: 75, 436, 1067,
8, 32,
```

```
int main(int argc, char **argv) (!!!)
```

```
#include <iostream>

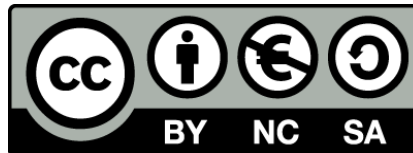
int main(int argc, char **argv)
{
 for(int i = 0; i<argc; i++)
 cout << "argument #" << i << " is '" << argv[i] << "'\n";
}
```

```
> ./test this is just a bunch of arguments

argument #0 is './test'
argument #1 is 'this'
argument #2 is 'is'
argument #3 is 'just'
argument #4 is 'a'
argument #5 is 'bunch'
argument #6 is 'of'
argument #7 is 'arguments'
```



# ΤΕΛΟΣ ΕΝΟΤΗΤΑΣ



# Σημειώματα



# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Δυτικής Αττικής, Κλειώ Σγουροπούλου 2020. Κλειώ Σγουροπούλου. «Αντικειμενοστραφής Προγραμματισμός-Θ. Ενότητα 1: Περιήγηση στη C++». Έκδοση: 1.0. Αθήνα 2020. Διαθέσιμο από τη δικτυακή διεύθυνση: [eclass.uniwa.gr](http://eclass.uniwa.gr).

# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό. Οι όροι χρήσης των έργων τρίτων επεξηγούνται στη διαφάνεια «Επεξήγηση όρων χρήσης έργων τρίτων».

Τα έργα για τα οποία έχει ζητηθεί άδεια αναφέρονται στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

# Επεξήγηση όρων χρήσης έργων τρίτων

|                                      |                                                                                                                                                                                    |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ©                                    | Δεν επιτρέπεται η επαναχρησιμοποίηση του έργου, παρά μόνο εάν ζητηθεί εκ νέου άδεια από το δημιουργό.                                                                              |
| διαθέσιμο με άδεια CC-BY             | Επιτρέπεται η επαναχρησιμοποίηση του έργου και η δημιουργία παραγώγων αυτού με απλή αναφορά του δημιουργού.                                                                        |
| διαθέσιμο με άδεια CC-BY-SA          | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού, και διάθεση του έργου ή του παράγωγου αυτού με την ίδια άδεια.                                               |
| διαθέσιμο με άδεια CC-BY-ND          | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού.<br>Δεν επιτρέπεται η δημιουργία παραγώγων του έργου.                                                         |
| διαθέσιμο με άδεια CC-BY-NC          | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού.<br>Δεν επιτρέπεται η εμπορική χρήση του έργου.                                                               |
| διαθέσιμο με άδεια CC-BY-NC-SA       | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού και διάθεση του έργου ή του παράγωγου αυτού με την ίδια άδεια.<br>Δεν επιτρέπεται η εμπορική χρήση του έργου. |
| διαθέσιμο με άδεια CC-BY-NC-ND       | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού.<br>Δεν επιτρέπεται η εμπορική χρήση του έργου και η δημιουργία παραγώγων του.                                |
| διαθέσιμο με άδεια CC0 Public Domain | Επιτρέπεται η επαναχρησιμοποίηση του έργου, η δημιουργία παραγώγων αυτού και η εμπορική του χρήση, χωρίς αναφορά του δημιουργού.                                                   |
| διαθέσιμο ως κοινό κτήμα             | Επιτρέπεται η επαναχρησιμοποίηση του έργου, η δημιουργία παραγώγων αυτού και η εμπορική του χρήση, χωρίς αναφορά του δημιουργού.                                                   |
| χωρίς σήμανση                        | Συνήθως δεν επιτρέπεται η επαναχρησιμοποίηση του έργου.                                                                                                                            |

# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.