

## ΔΕΙΚΤΕΣ ΚΑΙ ΣΥΝΑΡΤΗΣΕΙΣ

### ΚΛΗΣΗ ΜΕ ΤΙΜΗ

Σε αυτό το πρόγραμμα η συνάρτηση *athroisma* μεταβιβάζει αντίγραφα των τιμών των μεταβλητών *a* και *b* στις τοπικές μεταβλητές *x* και *y* για να υπολογίσει το άθροισμα. Οι μεταβλητές *a* και *b* δεν επηρεάζονται από την συνάρτηση.

```
#include<iostream>
int athroisma(int,int);
int main()
{
    int number1, number2, sum;
    cout << "Give First Number: "; cin >> number1;
    cout << "Give Second Number: "; cin >> number2;
    sum = athroisma(number1, number2);
    cout << "====Result====" << endl;
    cout << "\t" << number1 << "+" << number2 << "=" << sum << endl;
    return 0;
}
int athroisma(int x,int y){
    return x + y;
}
```

### ΚΛΗΣΗ ΜΕ ΑΝΑΦΟΡΑ

Σε αυτό το πρόγραμμα η δήλωση `int &i` σημαίνει ότι η παράμετρος *i* αναφέρεται στην μεταβλητή *sum* και αντιστοίχως η δήλωση `int &j` σημαίνει ότι η παράμετρος *j* στην μεταβλητή *diff*. Ουσιαστικά το *i* είναι ψευδώνυμο του *sum* και το *j* ψευδώνυμο του *diff*. Οποιαδήποτε αλλαγή υπάρξει στα ψευδώνυμα, αυτομάτως γίνεται και στις μεταβλητές

```
#include<iostream>
void sumDiff(int &, int &, int, int);
int main()
{
    int number1, number2, sum, diff;
    cout << "Give First Number: "; cin >> number1;
    cout << "Give Second Number: "; cin >> number2;
    sumDiff(sum, diff, number1, number2);
    cout << "====Result====" << endl;
    cout << "\t" << number1 << "+" << number2 << "=" << sum << endl;
    cout << "\t" << number1 << "-" << number2 << "=" << diff << endl;
    return 0;
}
void sumDiff(int &i, int &j, int x, int y){
    i = x + y;
    j = x - y;
}
```

### ΚΛΗΣΗ ΜΕ ΔΕΙΚΤΕΣ

Σε αυτό το πρόγραμμα κατά την κλήση της συνάρτησης μεταβιβάζονται οι διευθύνσεις των μεταβλητών *sum* και *diff* στους δείκτες *\*i* και *\*j*. Όταν γράψουμε λοιπόν `*i=x+y`; και `*j=x-y`; έχουμε πρόσβαση με τους δείκτες *\*i* και *\*j* στο περιεχόμενο των μεταβλητών αυτών.

```
#include<iostream>
void sumDiff(int *, int *, int, int);
int main()
{
    int number1, number2, sum, diff;
    cout << "Give First Number: "; cin >> number1;
    cout << "Give Second Number: "; cin >> number2;
```

```

sumDiff(&sum, &diff, number1, number2);
cout << "====Result====" << endl;
cout << "\t" << number1 << "+" << number2 << "=" << sum << endl;
cout << "\t" << number1 << "-" << number2 << "=" << diff << endl;
return 0;
}
void sumDiff(int *i, int *j, int x, int y){
    *i = x + y;
    *j = x - y;
}

```

#### ΔΕΙΚΤΕΣ ΚΑΙ ΣΥΜΒΟΛΟΣΕΙΡΕΣ

Στο παρακάτω πρόγραμμα ορίζουμε την συμβολοσειρά "VISUAL C++", τυπώνουμε έναν έναν τους χαρακτήρες της, μετά τυπώνουμε τους χαρακτήρες με ανάποδη σειρά και τέλος παίρνουμε τους τρεις τελευταίους χαρακτήρες και τους μεταφέρουμε σε μία άλλη συμβολοσειρά με όνομα CPP.

```

#include<iostream>
int main()
{
    int i;
    char array[] = "VISUAL C++";
    char cpp[3];
    char *ptr, *ptr1; //δείκτες τύπου char
    ptr = &array[0]; //ο δείκτης δείχνει στο 1ο στοιχείο του πίνακα array
    cout << "\n\nΕΚΤΥΠΩΣΗ ΤΗΣ ΣΥΜΒΟΛΟΣΕΙΡΑΣ " << endl;
    cout << "=====" << endl;
    for(i=0; i<10; i++)
    {
        cout << " \t array[" << i << "]= " << *ptr << endl;
        ptr++; //ο δείκτης δείχνει στο επόμενο στοιχείο του πίνακα array
    }
    ptr = &array[0] + 9; //ο δείκτης δείχνει στο 10ο στοιχείο του πίνακα array
    cout << "\n\nΕΚΤΥΠΩΣΗ ΤΗΣ ΣΥΜΒΟΛΟΣΕΙΡΑΣ " ;
    cout << "ΑΠΟ ΤΟ ΤΕΛΟΣ " << endl;
    cout << "=====" << endl ;
    for(i=9; i>=0; i--)
    {
        cout << " \t array[" << i << "]= " << *ptr << endl;
        ptr--; //ο δείκτης δείχνει στο προηγούμενο στοιχείο του πίνακα array
    }
    ptr = &array[7]; //ο δείκτης δείχνει στο 8ο στοιχείο του πίνακα array
    ptr1 = &cpp[0]; //ο δείκτης δείχνει στο 1ο στοιχείο του πίνακα cpp
    for(i=0; i<3; i++)
    {
        *ptr1 = *ptr; //απόδοση της τιμής του ptr στον ptr1
        ptr++; //ο δείκτης δείχνει στο επόμενο στοιχείο του πίνακα array
        ptr1++; //ο δείκτης δείχνει στο επόμενο στοιχείο του πίνακα cpp
    }
    cout << "\n\nΟΙ 3 ΤΕΛΕΥΤΑΙΟΙ ΧΑΡΑΚΤΗΡΕΣ " << endl;
    cout << "=====" << endl ;
    cout << "\t";
    for(i=0; i<3; i++)
        cout << cpp[i];
    cout << endl;
    return 0;
}

```

## ΔΕΙΚΤΕΣ ΚΑΙ ΑΝΤΙΚΕΙΜΕΝΑ

Στο παρακάτω πρόγραμμα δηλώνουμε έναν δείκτη **S** τύπου **Student** ο οποίος δείχνει σε ένα ανώνυμο αντικείμενο της κλάσης και χειρίζεται τις μεθόδους που περιέχει η κλάση.

```
#include<iostream>
#include"Student.h"
using namespace std;

int main()
{
    Student *S; //δείκτης της κλάσης
    S = new Student; //δείκτης δείχνει σε ανώνυμο αντικείμενο
    cout << "===Get Informations===" << endl;
    S->fillInfo(); //κλήση μεθόδου από τον δείκτη
    cout << "\n===Print Informations===" << endl;
    S->showInfo();
    return 0;
}
```

## ΠΙΝΑΚΑΣ ΑΝΤΙΚΕΙΜΕΝΩΝ ΜΕ ΔΕΙΚΤΕΣ

Στο παρακάτω πρόγραμμα φτιάχνουμε έναν πίνακα δεικτών οι οποίοι χειρίζονται αντικείμενα της κλάσης Student την οποία συμπεριλαμβάνουμε στο πρόγραμμα ως αρχείο επικεφαλίδας (Student.h). Κατόπιν εισάγουμε φοιτητές και κάνουμε εκτύπωση όλων των φοιτητών ή φοιτητών συγκεκριμένου τμήματος.

```
#include<iostream>
#include"Student.h"
using namespace std;

int main()
{
    Student *S[100]; //πίνακας δεικτών της κλάσης
    int i = 0, j, k, info, cont, depid;
    char choice;
    bool found;
    cout << "\n===Data Entry===" << endl;

    do{
        S[i] = new Student; //νέο αντικείμενο στο οποίο δείχνει ο δείκτης
        S[i] -> fillInfo();
        i++;
        cout << "\nMore? Y to continue OR N to stop insert: "; cin >> choice;
        cout << endl;
    }while (choice == 'Y' || choice == 'y');

    do{
        cout << "Informations for all students press 1, ";
        cout << "Informations for the students of a specific department press
2: ";

        cin >> info;
        if(info == 1){ //εκτύπωση όλων των φοιτητών
            cout << "\n===All Imported Students===" << endl;
            for (j = 0; j < i; j++){
                S[j] -> showInfo();
                cout << endl;
            }
        }
    }
```

```

    }
    if(info == 2){
        do{ //επιλογή φοιτητών συγκεκριμένου τμήματος
            cout << "Choose Department: "; cin >> depid;
            k = 0;
            found == false;
            while(k < i || found == false){
                if(S[k]->getDepartment() == depid) found == true;
                else k++;
            }
            if(found == true){
                cout << "\n===Students studying in the department " <<
depid << " are===" << endl;
                S[k]->showInfo(); //εκτύπωση στοιχείων φοιτητή
            }
            else
                cout << "The Department does not exist" << endl;

                cout << "To continue press Y: "; cin >> choice;
            }while(choice == 'Y' || choice == 'y');
        }
        cout << "\nTo continue with Informations Press 1: "; cin >> cont;
    }while(cont == 1);
    cout << "\nThank you!! Bye-Bye!!!" << endl;
    return 0;
}

```