Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Αγρονόμων Τοπογράφων Μηχανικών

Εισαγωγή στην πληροφορική

Βασίλειος Βεσκούκης
Δρ. Ηλεκτρολόγος Μηχανικός &
Μηχανικός Υπολογιστών ΕΜΠ
v.vescoukis@cs.ntua.gr

Η γλώσσα προγραμματισμού C++
Δηλώσεις μεταβλητών και σταθερών, είσοδος-έξοδος από streams, σύγκριση

Προγραμματισμός

Κυριολεκτικά:
Η συγγραφή των εντολών ενός προγράμματος

Εντολές = πηγαίος κώδικας (program source code)

Συνήθης χρήση του όρου "προγραμματισμός"

- **Προγραμματισμός = Ολη η διαδικασία ανάπτυξης λογισμικού**
- **Προγραμματισμός = Σχεδίαση και γράψιμο προγράμματος (πηγαίου κώδικα)**

«Τεχνοτροπίες» προγραμματισμού

- **Spaghetti programming**
- **Δομημένος προγραμματισμός**
- **Αντικειμενοστρεφής προγραμματισμός (2ο εξάμηνο)**
- **Λογικός προγραμματισμός**

Μια "τεχνοτροπία προγραμματισμού" προσδιορίζεται από

- **τα εργαλεία προγραμματισμού και**
- **τον τρόπο χρήσης τους**

Προγραμματισμός

Η συγγραφή του πηγαίου κώδικα για κάποιες μονάδες προγράμματος οι οποίες, όταν εκτελούνται, παράγουν ένα επιθυμητό αποτέλεσμα.

Κανόνες που ακολουθούνται

- **Συντακτικοί (γλώσσα προγραμματισμού)**
- **Σημασιολογικοί (γλώσσα προγραμματισμού)**
- **Λογικοί (πρόβλημα)**

Οι κανόνες της γλώσσας προγραμματισμού αφορούν...

- **Τα σύμβολα, τις εντολές, τις δομές της γλώσσας**
- **Τον τρόπο χρήσης των ιδιαίτερων χαρακτηριστικών κάθε υλοποίησης της γλώσσας (Visual C++, Borland C++, DEV-C++, gcpp, κλπ)**
- **Τους γενικούς κανόνες και παραδοχές που αφορούν όλα τα προγράμματα**

Η «δυσκολία» εκμάθησης μιας γλώσσας προγραμματισμού εντοπίζεται

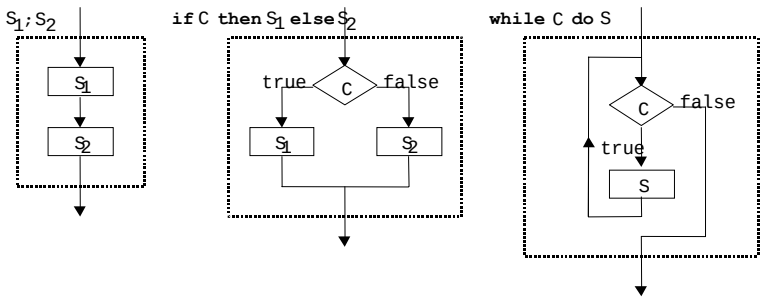
- **Στην απόσταση από τη φυσική γλώσσα του ανθρώπου**
- **Στους κανόνες σύνταξης οι οποίοι συμβάλλουν στον εμπλουτισμό της σημασιολογίας διότι οι Η/Υ δεν διαθέτουν ευφυία**

Δομημένος προγραμματισμός

Ο τρόπος συγγραφής πηγαίου κώδικα, σύμφωνα με τον οποίο η συντακτική δομή του προγράμματος βοηθάει στην κατανόηση της ροής ελέγχου αυτού, δηλαδή της σειράς εκτέλεσης των εντολών.

Τρεις δομές οργάνωσης προγράμματος (και τμημάτων αυτού):

- **Ακολουθία (`s1 ; s2`)**
- **Επιλογή (`if c then s1 else s2`)**
- **Επανάληψη (`while c do s`)**



Γράφοντας προγράμματα...

Μερικές χρήσιμες παρατηρήσεις

- Η λύση ενός προβλήματος είναι στην καλύτερη περίπτωση τόσο «καλή» όσο «καλός» είναι ο ορισμός του προβλήματος
- «Καλός» = σωστός και ακριβής
- Στη φάση του προγραμματισμού (πρέπει να) είναι ήδη γνωστό το «τι θα κάνει το πρόγραμμα»...
- έτσι ώστε να μας απασχολεί πώς το πρόγραμμα θα κάνει με *σωστό τρόπο* τα *σωστά πράγματα*

Ενα πρόγραμμα...

- είναι μια ακολουθία από εντολές μιας γλώσσας προγραμματισμού...
- οι οποίες οργανώνονται σε υποσύνολα που ονομάζονται «μονάδες προγράμματος»

Ενα (μεγαλύτερο) πρόγραμμα...

- είναι ένα σύνολο από μονάδες προγράμματος

Κάθε μονάδα προγράμματος

- είναι μια ακολουθία δομών μιας γλώσσας προγραμματισμού

Στοιχεία προγράμματος C++

Αναγνωριστικά ονόματα (identifiers)

- Μεταβλητές μνήμης
- Σταθερές

Τελεστές

- Αριθμητικοί
- Λογικοί
- Σύγκρισης

Εκφράσεις, εντολές

- Δεσμευμένες λέξεις
- Αριθμητικές και λογικές εκφράσεις
- Ελεγχος ροής προγράμματος

Συναρτήσεις

Κλάσεις

Σχόλια

Σχόλια: δεν "εκτελούνται", όμως βοηθούν πολύ!

```
1: #include <iostream.h>
2:
3: int main()
4: {
5:     /* this is a comment
6:     and it extends until the closing
7:     star-slash comment mark */
8:     cout << "Hello World!\n";
9:     // this comment ends at the end of the line
10:    cout << "That comment ended!\n";
11:
12:    // double slash comments can be alone on a line
13:    /* as can slash-star comments */
14:    return 0;
15: }
```

```
1: #include <iostream.h>
2:
3: int main()
4: {
5:     /* this is a comment
6:     and it extends until the closing
7:     star-slash comment mark */
8:     cout << "Hello World!\n";
9:     // this comment ends at the end of the line
10:    cout << "That comment ended!\n";
11:
12:    // double slash comments can be alone on a line
13:    /* as can slash-star comments */
14:    return 0;
15: }
```


Δεσμευμένες λέξεις της C++

asm **auto** bool **break** case catch **char** class
const const_cast **continue** **default** delete **do**
double dynamic_cast **else** **enum** explicit **extern**
false **float** **for** friend **goto** **if** inline **int** **long**
mutable namespace new operator private protected
public **register** reinterpret_cast **return** short
signed **sizeof** **static** static_cast **struct** **switch**
template this throw true try **typedef** typeid
typename **union** **unsigned** using virtual **void**
volatile wchar_t **while**

Οι λέξεις που σημειώνονται με **έντονα γράμματα** είναι δεσμευμένες λέξεις και στη **C**

Αναγνωριστικά ονόματα (identifiers)

Ενα αναγνωριστικό όνομα (identifier)

- μπορεί να αποτελείται από χαρακτήρες, αριθμούς και το underscore (_)
- πρέπει να ξεκινά από χαρακτήρα ή από το underscore (_)
- είναι case-sensitive (case <> Case <> CASE <> CAse κλπ)

Υπάρχουν προκαθορισμένα αναγνωριστικά ονόματα στη C++...

- cin, cout, κ.ά.
- Αυτά δεν επιτρέπεται να τα χρησιμοποιήσει ο προγραμματιστής...

...καθώς και αναγνωριστικά που ορίζονται από τον προγραμματιστή για να ονοματίσουν μεταβλητές μνήμης, συναρτήσεις, κλπ δομές προγράμματος

- `i, salary, CostPerHour, person1, first_name, Phone_Number`
- `1stName, miles per hour, calculate max, one+more, CalculateMin`

Η επιλογή των ονομάτων πρέπει να είναι κατάλληλη και να συμβάλλει στην κατανόηση του προγράμματος

- DailySalary = Hours * CostPerHour // κατανοητό
- ds = h * cph // σωστό, αλλά μη κατανοητό

Τύποι δεδομένων

Τα δεδομένα εισόδου, τα αποτελέσματα και όλα τα ενδιάμεσα υπολογιζόμενα δεδομένα αποθηκεύονται σε χώρο που δεσμεύεται στη μνήμη του Η/Υ

Ο τρόπος παράστασης των δεδομένων επιβάλλει τη διαφοροποίηση του χειρισμού τους από τον Η/Υ, ανάλογα με το είδος των εκάστοτε δεδομένων, διότι τελικά όλα είναι ακολουθίες από 0 και 1

- Ακέραιοι αριθμοί, Χαρακτήρες, Πραγματικοί αριθμοί, κλπ

ΤΥΠΟΣ (type)
Ενα σύνολο κανόνων χειρισμού και παράστασης δεδομένων από μια γλώσσα προγραμματισμού ονομάζεται τύπος δεδομένων (data type) [θυμάστε?]

Οι τύποι δεδομένων στη C++ διακρίνονται σε τρεις κατηγορίες

- Απλοί τύποι δεδομένων
- Δομημένοι τύποι δεδομένων
- Δείκτες

Τύποι δεδομένων

Απλοί τύποι δεδομένων

- **Ακέραιοι**
  - `char`,
  - `short, int, long,`
  - `bool,`
  - `unsigned char, unsigned short, unsigned int, unsigned long`
- **Κινητής υποδιαστολής (floating point)**
  - `float`
  - `double`
  - `long double`
- **Απαριθμητοί (enumerated)**

Δομημένοι τύποι

- **Πίνακες (tables, arrays)**
- **Δομές (structures)**
- **Συμβολοσειρές (strings)**

Δείκτες

- **Διευθύνσεις μεταβλητών μνήμης**

Μεταβλητές μνήμης και σταθερές

Μια μεταβλητή μνήμης (memory variable, variable) χρησιμοποιείται για την προσωρινή αποθήκευση δεδομένων που χρησιμοποιεί το πρόγραμμα που την δηλώνει

- **Διατύπωση:**
  - **Μια μεταβλητή μνήμης (memory variable) είναι κάποιου τύπου**
  - **Μια μεταβλητή μνήμης ανήκει σε κάποιον τύπο**
- **Δηλαδή:**
  - **Παριστάνεται και συμπεριφέρεται σύμφωνα με όσα καθορίζει ο τύπος στον οποίο ανήκει**

Μια σταθερά διατηρεί πάντα την ίδια τιμή

Παραδείγματα

```
const double pi=3.14;
const int arista=10;

int a;
float salary;
bool cleared;

int a=0;
double temperature, pressure, humidity;
```

Αριθμητικές εκφράσεις

Αριθμητικοί τελεστές

- **+, -, *, /**
  για τις τέσσερις βασικές πράξεις ακεραίων και αριθμών κινητής υποδιαστολής
- **%** (modulo) ακέραιο υπόλοιπο διαίρεσης ακεραίων αριθμών

Αριθμητικές εκφράσεις

- 3 + 4 = 7
- 6 * 4 / 2 = 12
- 10 – 3 * 2 = 4
- 5 * 5 – 4 * 6 = 1
- 5 / 2 = 2 (ακέραιοι)
- 5 % 2 = 1 (ακέραιοι)

Quiz!

- 4 % 6 =
- 25 % 26 =
- 12 / 5 =
- 3 * 4 * 2 – 4 * 6 + 10 % 3 =
- 5 * 9 + 3 % 4 + 2 =

Αριθμητικές εκφράσεις

Κινητής υποδιαστολής

- 5.0 + 3.0 = 8.0 3.3 * 3.0 = 9.9
- 15.5 – 5.4 = 10.1 10.0 / 2.5 + 1.0 = 5

Προτεραιότητες

- Οι *, /, % έχουν όλοι την ίδια προτεραιότητα
- Οι +, - έχουν μικρότερη προτεραιότητα
- Σε εκφράσεις με τελεστές ίδιας προτεραιότητας, η αποτίμηση γίνεται από αριστερά προς τα δεξιά
- Μπορούν να χρησιμοποιούνται παρενθέσεις για να καθιστούν σαφή την επιθυμητή προτεραιότητα

Παραδείγματα

- 3 * 7 – 6 + 2 * 5 / 4 + 6 =
  21 – 6 + (10 / 4) + 6 =
  15 + 2 + 6 =
  23

  (3 * 7) - 6 + ((2 * 5) / 4) + 6 =
  21 – 6 + 2 + 6 =
  17 + 6 =

Αριθμητικές εκφράσεις

Χαρακτηρισμός εκφράσεων ως προς τον τύπο

- **Ακέραιες: όλα τα τελούμενα είναι ακέραιου τύπου**
  - **2 + 3 * 5, 3 + x - y / 7, x + 2 * (y - z) + 18**
- **Κινητής υποδιαστολής ή δεκαδικές: όλα τα τελούμενα είναι αριθμοί κινητής υποδιαστολής**
  - **12.8 * 17.5 - 34.50, x * 10.5 + y - 16.2**
- **Μικτές: κάποια από τα τελούμενα της ίδιας έκφρασης είναι ακέραιοι και κάποια είναι αριθμοί κινητής υποδιαστολής**
  - **6 / 4 + 3.9, 5.4 * 2 – 13.6 + 18 / 2**

Αποτίμηση μικτών εκφράσεων

- **Αν ένας τελεστής έχει τελούμενα ιδίου τύπου, αποτιμάται ως τον τύπο εκείνο**
- **Αν υπάρχουν τελούμενα διαφορετικών τύπων, το ακέραιο μετατρέπεται σε κινητής υποδιαστολής και η αποτίμηση έχει ως αποτέλεσμα αριθμό κινητής υποδιαστολής**
- **Οι κανόνες προτεραιότητας ισχύουν ως αναφέρθηκαν**

Παραδείγματα αποτίμησης εκφράσεων

3 / 2 + 5.0

= 1 + 5.0	**(3 / 2 = 1)**
= 6.0	**(1 + 5.0 = 1.0 + 5.0= 6.0)**

15.6 / 2 + 5

= 7.8 + 5	**(15.6 / 2 = 15.6 / 2.0 = 7.8)**
= 12.8	**(7.8 + 5.0 = 12.8)**

4 * 3 + 7 / 5 – 25.6

= 12 + 7 / 5 – 25.6	**(4 * 3 = 12)**
= 12 + 1 – 25.6	**(7 / 5 = 1)**
= 13 – 25.6	**(12 + 1 = 13)**
= -12.6	**(13 – 25.6 = 13.0 – 25.6 = - 12.6)**

Περί ονοματολογίας μεταβλητών και σταθερών

ΠΡΟΣΟΧΗ ΣΤΑ ΟΝΟΜΑΤΑ ΠΟΥ ΟΡΙΖΟΥΜΕ (ΜΤΒΛ, ΣΤΑΘΕΡΕΣ, ΣΥΝΑΡΤΗΣΕΙΣ)!

```
x=3.14*r*r
```

ή μήπως

```
area = pi * radius * radius
```

```
_ps = wd * 50 - 0.12 * wd * 50 - 0.05 * wd * 50
```

ή καλύτερα...

```
PayableSal = WorkedDays * DailyWage * (1 - TaxRate - InsuranceRate)
```

Τα ονόματα των μεταβλητών που χρησιμοποιούμε πρέπει να μας βοηθούν στην κατανόηση του προγράμματος

Μετατροπές τύπων (type casting) στη C++

Τελεστής μετατροπής τύπου (cast operator):

```
static_cast<dataTypeName>(expression)
```

Παραδείγματα

```
1. static_cast<int>(7.9) = 7
2. static_cast<int>(3.3) = 3
3. static_cast<double>(25) = 25.0
4. static_cast<double>(5+3) = static_cast<double>(8) = 8.0
5. static_cast<double>(15)/2      = 15.0/2 = 15.0/2.0 = 7.5
6. static_cast<int>(7.8 + static_cast<double>(15)/2)=
        static_cast<int>(7.8 + 7.5) = static_cast<int>(15.3) = 15

[x = 15  y = 23  z = 3.75]
7. static_cast<double>(y / x) + z = 4.75
8. static_cast<double>(y) / x + z = 5.28333
```

Είσοδος (input)

2 βήματα

- **Δέσμευση μνήμης: δήλωση μεταβλητών και σταθερών**
- **Χρήση εντολών που τοποθετούν/μεταβάλλουν δεδομένα που αποθηκεύονται στην (ήδη) δεσμευμένη μνήμη**

1. Δέσμευση μνήμης

- `const dataType [identifier] = [value];`
- `dataType [identifier], [identifier], . . .;`

Παραδείγματα

- `const double pi=3.14;`
- `double temperature, pressure, humidity;`

2. Χρήση μνήμης

- **Ανάθεση τιμής σε μεταβλητή [= : assignment operator]**
  - `variable = expression;`
- **Ανάγνωση τιμής από πηγή εισόδου [>> : extraction operator]**
  - `cin>>variable>>variable. . .;`

Εντολή ανάθεσης

Παραδείγματα

```
int I, J;
double sale;
char first;
string str;

I = 4;
J = 4 * 5 - 11;
sale = 0.02 * 1000;
first = 'D';
str = "It is a sunny day. ";
```

Εξέλιξη τιμών της ίδιας μεταβλητής

```
int I, J=1;
double a=3, b=4, c=1, discriminator;

I=6;I=I+1;I=J%4;I=I*I;
J=4%3+4*I;
J=J*I;

discriminator=b*b-4*a*c
```

Ανάγνωση τιμής από πηγή εισόδου

cin>>[variable]

- **cin**: **c**onsole **in**put,
  δηλώνεται στο <iostream>
- **>>**: extraction operator

Χρήση

- Δήλωση της βιβλιοθήκης iostream
  [#include <iostream>]
- Δήλωση μεταβλητών μνήμης
  [πχ: double salary;
  int days;]
- Προαιρετικά: αρχικοποίηση τιμών μεταβλητών μνήμης
  [πχ: days=0 ;]
- Ανάγνωση τιμών από τη μονάδα εισόδου (πληκτρολόγιο)

```
#include <iostream>;

using namespace std;

int main()
{
    // ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ ΜΝΗΜΗΣ
    double dailywage;
    int days;
    double salary;

    /* ΕΙΣΟΔΟΣ ΜΕ ΑΝΑΓΝΩΣΗ
       ΑΠΟ ΤΟ ΠΛΗΚΤΡΟΛΟΓΙΟ */
    cin>>days;
    cin>>dailywage;

    // ΥΠΟΛΟΓΙΣΜΟΙ
    salary=days*dailywage;

    //ΕΞΟΔΟΣ
    cout<<salary;

    return 0;
}
```


Εξοδος στην οθόνη

cout<<[έκφραση] ή [χειριστής εξόδου]

- **cout**: **c**onsole **out**put,
  δηλώνεται στο iostream
- **<<**: insertion operator

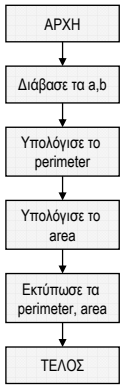
Χρήση

- Δήλωση της βιβλιοθήκης iostream
  [#include <iostream>]
- Δήλωση και χρήση μεταβλητών μνήμης
  [πχ: double dailywage, salary;
  int days;
  cin<<days;
  cin<<dailywage;
  salary=dailywage*days;]
- Εγγραφή τιμών στη μονάδα εξόδου
  endl: "κατεβάζει" μια γραμμή

```
#include <iostream>;

using namespace std;

int main()
{
    // ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ ΜΝΗΜΗΣ
    double dailywage;
    int days;
    double salary;

    /* ΕΙΣΟΔΟΣ ΜΕ ΑΝΑΓΝΩΣΗ
       ΑΠΟ ΤΟ ΠΛΗΚΤΡΟΛΟΓΙΟ */
    cin>>days;
    cin>>dailywage;

    // ΥΠΟΛΟΓΙΣΜΟΙ
    salary=days*dailywage;

    //ΕΞΟΔΟΣ
    cout<<salary;

    return 0;
}
```

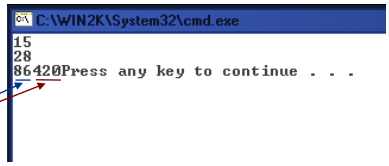
Παράδειγμα 1, μέρος 2α

```
#include <iostream>;

using namespace std;


int main()
{
    // ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ
    float a, b, perimeter, area;

    // ΕΙΣΟΔΟΣ: "ΔΙΑΒΑΣΕ ΤΑ A,B"
    cin>>a;
    cin>>b;

    // ΕΠΕΞΕΡΓΑΣΙΑ: "ΥΠΟΛΟΓΙΣΕ ΤΟ perimeter"
    perimeter=2*(a+b);

    // ΕΠΕΞΕΡΓΑΣΙΑ: "ΥΠΟΛΟΓΙΣΕ ΤΟ area"
    area=a*b;

    // ΕΞΟΔΟΣ: "ΤΥΠΩΣΕ ΤΑ perimeter, area"
    cout<<perimeter<<endl;
    cout<<area<<endl;

    // ΤΕΛΟΣ
    return 0;
}
```



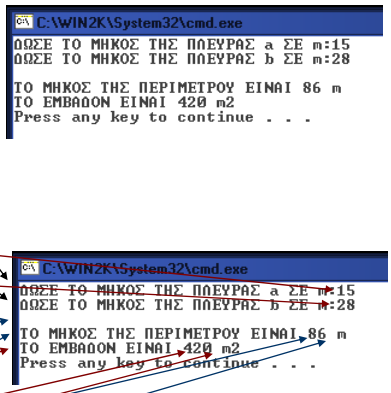
Παράδειγμα 1, μέρος 2β

```
#include <iostream>

using namespace std;

int main()
{
    // ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ ΜΝΗΜΗΣ
    float a, b, perimeter, area;

    // ΕΙΣΟΔΟΣ
    cin>>a;
    cin>>b;


    // ΥΠΟΛΟΓΙΣΜΟΙ
    perimeter=2*(a+b);
    area=a*b;

    // ΕΞΟΔΟΣ
    cout<<perimeter;
    cout<<area;

    // ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ
    return 0;
}
```



Παράδειγμα 1, μέρος 2γ

```
#include <iostream>

using namespace std;

int main()
{
    // ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ ΜΝΗΜΗΣ
    float a, b, perimeter, area;


    // ΕΙΣΟΔΟΣ
    cout<<"ΔΩΣΕ ΤΟ ΜΗΚΟΣ ΤΗΣ ΠΛΕΥΡΑΣ a ΣΕ m:";
    cin>>a;

    cout<<"ΔΩΣΕ ΤΟ ΜΗΚΟΣ ΤΗΣ ΠΛΕΥΡΑΣ b ΣΕ m:";
    cin>>b;


    // ΥΠΟΛΟΓΙΣΜΟΙ
    perimeter=2*(a+b);
    area=a*b;

    // ΕΞΟΔΟΣ
    cout<<"\n";
    cout<<"ΤΟ ΜΗΚΟΣ ΤΗΣ ΠΕΡΙΜΕΤΡΟΥ ΕΙΝΑΙ "<<perimeter<<" m\n";
    cout<<"ΤΟ ΕΜΒΑΔΟΝ ΕΙΝΑΙ "<<area<<" m2\n";

    // ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ
    return 0;
}
```



Παράδειγμα εισόδου-εξόδου

Τι τυπώνει το παρακάτω πρόγραμμα;

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      float a=10, b, c;
7
8      cin>>a;
9      cout<<a;
10     cout<<a<<b<<c;
11     cout<<"\n"<<a<<"\n"<<b<<"\n"<<c<<"\n";
12
13     a=100.0; b=a+10; c=b-100;
14     cout<<"a="<<a<<", b="<<b<<", "<<"c="<<c;
15
16     cout<<"the value of a is"<<b<<"\n";
17
18     cout<<"the sum of a and b is"<<a+b;
19     cout<<"where the sum of b and c is"<<b+c;
20
21     cout<<"\n\nPlease enter an integer value";
22     cin>>a;
23
24     cout<<"You have just entered "<<a;
25 }
```

Παράδειγμα εισόδου-εξόδου

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      float a=10, b, c;
7
8      cin>>a;
9      cout<<a;
10     cout<<a<<b<<c;
11     cout<<"\n"<<a<<"\n"<<b<<"\n"<<c<<"\n";
12
13     a=100.0; b=a+10; c=b-100;
14     cout<<"a="<<a<<", b="<<b<<", "<<"c="<<c;
15
16     cout<<"the value of a is"<<b<<"\n";
17
18     cout<<"the sum of a and b is"<<a+b;
19     cout<<"where the sum of b and c is"<<b+c;
20
21     cout<<"\n\nPlease enter an integer value";
22     cin>>a;
23
24     cout<<"You have just entered "<<a;
25 }
```

```
C:\WIN2K\System32\cmd.exe
999
9999993.58732e-431.4013e-45
999
3.58732e-43
1.4013e-45
a=100, b=110, c=10the value of a is110
the sum of a and b is210where the sum of b and c is120

Please enter an integer value500
You have just entered 500Press any key to continue . .
```


Παράδειγμα εισόδου-εξόδου

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      float a=10, b, c;
7
8      cin>>a;
9      cout<<a;
10     cout<<a<<b<<c;
11     cout<<"\n"<<a<<"\n"<<b<<"\n"<<c<<"\n";
12
13     a=100.0; b=a+10; c=b-100;
14     cout<<"a="<<a<<", b="<<b<<", "<<"c="<<c;
15
16     cout<<"the value of a is"<<b<<"\n";
17
18     cout<<"the sum of a and b is"<<a+b;
19     cout<<"where the sum of b and c is"<<b+c;
20
21     cout<<"\n\nPlease enter an integer value";
22     cin>>a;
23
24     cout<<"You have just entered "<<a;
25 }
```

```
C:\WIN2K\System32\cmd.exe
999
9999993.58732e-431.4013e-45
999
3.58732e-43
1.4013e-45
a=100, b=110, c=10the value of a is110
the sum of a and b is210where the sum of b and c is120

Please enter an integer value500
You have just entered 500Press any key to continue . .
```

Δομές ελέγχου 0: ακολουθιακή εκτέλεση

Μια εντολή μπορεί να είναι:

- **ΑΠΛΗ: μια ανάθεση τιμής, κλίση συνάρτησης, είσοδος-έξοδος, κ.ά.**
- **ΣΥΝΘΕΤΗ: μια ακολουθία από απλές εντολές που τοποθετούνται μέσα σε άγκιστρα { } και ονομάζονται BLOCK εντολών**

ΑΠΛΕΣ ΕΝΤΟΛΕΣ

- **x=5;**
- **cin<<y;**
- **a=min(x,y);**

BLOCK εντολών

- **{**
  x=5;
  cin<<y;
  a=min(x,y);
  }

Δομές ελέγχου 0: ακολουθιακή εκτέλεση

Χρησιμοποιούμε block εντολών...

- **Εκεί που θέλουμε να εκτελέσουμε ένα σύνολο εντολών αν ισχύει μια συνθήκη**
- **Εκεί που θέλουμε να εκτελέσουμε ένα σύνολο εντολών επαναληπτικά**

ΔΗΛΑΔΗ μέσα σε δομές σύγκρισης και επανάληψης

Παράδειγμα

- **Αν το a έχει την τιμή 0 τότε τύπωσε ότι η εξίσωση είναι αδύνατη και δώσε στη μεταβλητή root την τιμή 0**
- **ΛΑΘΟΣ**

```
if (a==0)
    cout<<"No roots in R";
    root=0;
```

- **ΣΩΣΤΟ**

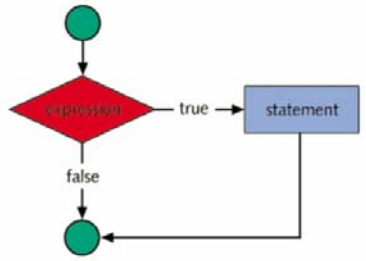
```
if (a==0)
{   cout<<"No roots in R";
    root=0;
}
```

Δομές ελέγχου 1: εκτέλεση υπό συνθήκη

ΜΟΡΦΗ 1

```
if (λογική συνθήκη)
   [εντολή];
```

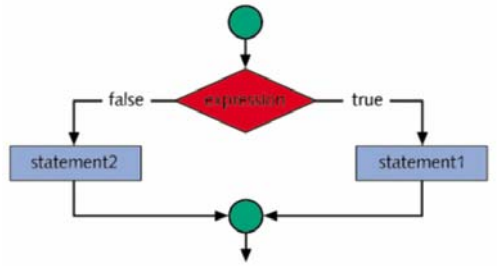
```
if (salary>50)
  T=0.08;
```



ΜΟΡΦΗ 2

```
if (λογική συνθήκη)
   [εντολή 1];
else
   [εντολή 2];
```

```
if (salary>50)
  T=0.08;
else
  T=0.05;
```



Λογικές εκφράσεις (1)

ΣΧΕΣΙΑΚΟΙ ΤΕΛΕΣΤΕΣ (relational operators)

- Επιτρέπουν την πραγματοποίηση συγκρίσεων μεταξύ τιμών
- ==, !=, <, <=, >, >=

ΑΠΛΕΣ ΛΟΓΙΚΕΣ ΕΚΦΡΑΣΕΙΣ

- Συγκρίσεις 2 τιμών που μπορούν να ισχύουν (true) ή όχι (false)
  - 10==10 (true)
  - 10<=10 (true)
  - 5>32 (false)
  - 4!=3 (true)
  - 5<=10 (true)
  - 6!=6 (false)
  - 2+3<10-4 (true)
  - 6.8+3.1==7.3+2.6 (συνήθως true)
  - 'A'<'B' (true)
  - "Παπαβασιλείου">"Παπαγεωργίου" (false)

ΠΡΟΣΟΧΗ ΣΤΗ ΣΥΓΧΥΣΗ ΤΟΥ == ΜΕ ΤΟ =

Λογικές εκφράσεις (2)

ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ (logical operators)

- Επιτρέπουν τη δημιουργία σύνθετων λογικών εκφράσεων από απλές
- ! (not), && (and), || (or)
- Ο τελεστής ! δέχεται μόνο ένα όρισμα, οι && και || δέχονται δύο

ΣΥΝΘΕΤΕΣ ΛΟΓΙΚΕΣ ΕΚΦΡΑΣΕΙΣ

- Συσχέτιση περισσοτέρων από μία λογικών εκφράσεων με χρήση λογικών τελεστών
- Επιστρέφουν `true` ή `false` ανάλογα με την αποτίμηση των μερών τους

ΠΑΡΑΔΕΙΓΜΑΤΑ

- (10>5) && (2>0) — true
- (100<200) || (3>10) — true
- !(1>0) — false
- (100>90) && !(2>3) — true
- ((2>1) || (5>6)) && (10>20) — false
- (2>1) || ((5>6) && (10>20)) — true

Λογικές εκφράσεις (3)

ΠΙΝΑΚΕΣ ΑΛΗΘΕΙΑΣ

Expression	!(Expression)
true (nonzero)	false (0)
false (0)	true (1)

Expression1	Expression2	Expression1 && Expression2
true (nonzero)	true (nonzero)	true (1)
true (nonzero)	false (0)	false (0)
false (0)	true (nonzero)	false (0)
false (0)	false (0)	false (0)

Expression1	Expression2	Expression1 || Expression2
true (nonzero)	true (nonzero)	true (1)
true (nonzero)	false (0)	true (1)
false (0)	true (nonzero)	true (1)
false (0)	false (0)	false (0)

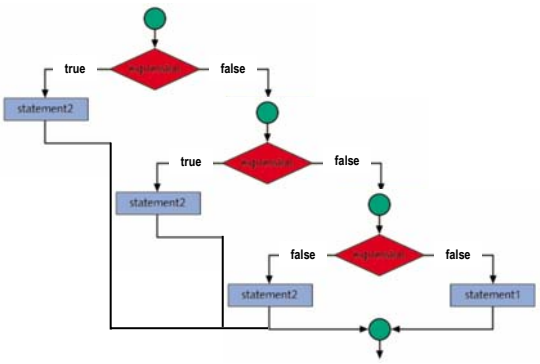
Παραδείγματα λογικών εκφράσεων

```
int a,b,c;
a=100; b=200; c=300;
                              // Να δοθούν οι τιμές
if (a>b)                      // true/false
  c=400;                      //      a:      b:      c:
if ((a>b) && (c>b))           // true/false
  a=350;                      //      a:      b:      c:
if ((a<b) || (b>c))           // true/false
  b=500;                      //      a:      b:      c:

//να αποτιμηθούν ως αληθείς ή ψευδείς οι παρακάτω εκφράσεις
a=1; b=2; c=3;
(a==2) || (b>c)
(a>5) || (b>a) && (c<100)
((a>5) || (b>a)) && (c<100)
(a>b) && (b>c) && (c>1)
(a>b)) || (b>c) || (c>1)
(c%b>=a) && (a*b*c<10)
(a<c-b) || ((a>0) || ((c<=2) && (c>=0)))
```


Δομές ελέγχου 1: εκτέλεση υπό συνθήκη

ΜΟΡΦΗ 3

```
if (λογική συνθήκη1)
   [εντολή 1];
else if (λογική συνθήκη2)
   [εντολή 2];
else if (λογική συνθήκη3)
   [εντολή 3];
else if (λογική συνθήκη4)
   [εντολή 4];
...
```




```
if (salary>50)
  T=0.08;
else if (salary>40)
  T=0.07;
else if (salary>30)
  T=0.06;
```

Προσοχή στη λογική και στη σειρά των λογικών εκφράσεων!

Παραδείγματα λογικών εκφράσεων

```
#include <iostream>
using namespace std;

main()
{
  int given_age;
  const int true_age=47;
  cout<<"Ποια είναι η ηλικία σας;";
  cin>>given_age;

if (true_age-given_age>=10)
  cout<<"Λυπάμαι, αλλά ψεύδεστε πολύ";
else if (true_age-given_age>=5)
  cout<<"Ψεύδεστε, αλλά λίγο";
else if (true_age==given_age)
  cout<<"Συγχαρητήρια για την ειλικρίνεια!";
else if (true_age<given_age)
  cout<<"Βιάζεστε...";
}
```


ΠΡΟΣΟΧΗ!

Ο πηγαίος κώδικας

```
int a=1;b;
if (a==2)
  b=a*100;
else
  b=0;
```

έχει ως αποτέλεσμα το αναμενόμενο, δηλαδή η μεταβλητή μνήμης b παίρνει την τιμή 0.

τα `if (a=b)`
και `if (a==b)`

ΚΑΝΟΥΝ ΔΙΑΦΟΡΕΤΙΚΑ ΠΡΑΓΜΑΤΑ!!!

ΟΜΩΣ: Ο πηγαίος κώδικας

```
if (a=2)
  b=a*100;
else
  b=0;
```

ισοδυναμεί με

```
a=2;
if (a!=0)
  b=a*100;
else
  b=0;
```

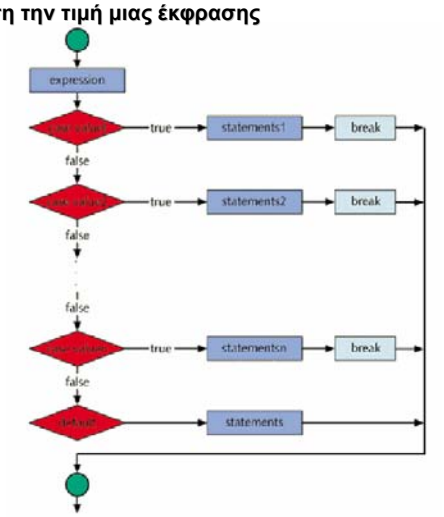
Και έχει ως αποτέλεσμα το b να πάρει την τιμή 200

Διότι χρησιμοποιήσαμε τον τελεστή ανάθεσης (=) αντί για τον τελεστή σύγκρισης (==)

Δομές ελέγχου 1: εκτέλεση υπό συνθήκη

Η ΔΟΜΗ `switch`
Επιλογή μονοπατιού εκτέλεσης με βάση την τιμή μιας έκφρασης

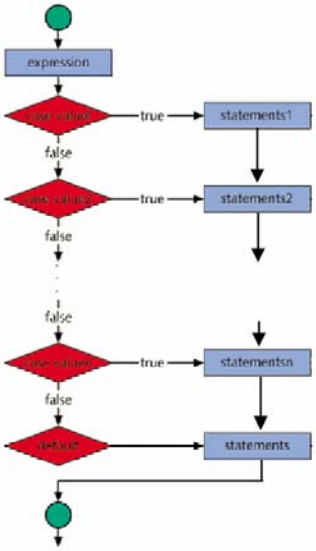
```
switch(expression)
{
case value1: statements1;
                    break;
case value2: statements2;
                    break;
       .
       .
       .
case valueN: statementsN;
                    break;
default: statements;
}
```



Προσοχή στη χρήση του switch!

Χωρίς break, εκτελούνται όλες οι εντολές

```
switch(expression)
{
case value1: statements1;

case value2: statements2;

       .
       .
       .
case valueN: statementsN;

default: statements;
}
```



Παράδειγμα

```
/* ------------------------------------------------------
   C++
   Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ switch (1)
   ------------------------------------------------------ */

#include <iostream>
using namespace std;

int main()
{
    int DayOfWeek=0;

    cout<<"ΕΠΕΛΕΞΕ ΤΗΝ ΗΜΕΡΑ ΤΗΣ ΕΒΔΟΜΑΔΑΣ (1-7):";
    cin>>DayOfWeek;

    cout<<"\nΗ "<<DayOfWeek<<"η ΗΜΕΡΑ ΤΗΣ ΕΒΔΟΜΑΔΑΣ ΕΙΝΑΙ Η ";

    switch (DayOfWeek)
    {
        case 1: cout<<"ΔΕΥΤΕΡΑ";
                break;
        case 2: cout<<"ΤΡΙΤΗ";
                break;
        case 3: cout<<"ΤΕΤΑΡΤΗ";
                break;
        case 4: cout<<"ΠΕΜΠΤΗ";
                break;
        case 5: cout<<"ΠΑΡΑΣΚΕΥΗ";
                break;
        case 6: cout<<"ΣΑΒΒΑΤΟ";
                break;
        case 7: cout<<"ΚΥΡΙΑΚΗ";
                break;
        default: cout<<"Πρέπει να δώσεις 1-7";
    }
    cout<<"\n";
}
```

```
C:\WIN2K\System32\cmd.exe
ΕΠΕΛΕΞΕ ΤΗΝ ΗΜΕΡΑ ΤΗΣ ΕΒΔΟΜΑΔΑΣ (1-7):5

Η 5η ΗΜΕΡΑ ΤΗΣ ΕΒΔΟΜΑΔΑΣ ΕΙΝΑΙ Η ΠΑΡΑΣΚΕΥΗ
Press any key to continue . . .
```


Παράδειγμα

```
#include <iostream>
using namespace std;

int main()
{
    int DayOfWeek=0;

    cout<<"ΕΠΕΛΕΞΕ ΤΗΝ ΗΜΕΡΑ ΤΗΣ ΕΒΔΟΜΑΔΑΣ (1-7):";
    cin>>DayOfWeek;

    cout<<"\nΗ "<<DayOfWeek<<"η ΗΜΕΡΑ ΤΗΣ ΕΒΔΟΜΑΔΑΣ ΕΙΝΑΙ Η ";

    switch (DayOfWeek)
    {
        case 1: cout<<"ΔΕΥΤΕΡΑ";

        case 2: cout<<"ΤΡΙΤΗ";

        case 3: cout<<"ΤΕΤΑΡΤΗ";

        case 4: cout<<"ΠΕΜΠΤΗ";

        case 5: cout<<"ΠΑΡΑΣΚΕΥΗ";

        case 6: cout<<"ΣΑΒΒΑΤΟ";

        case 7: cout<<"ΚΥΡΙΑΚΗ";

        default: cout<<"Πρέπει να δώσεις 1-7";
    }
    cout<<"\n";
}
```

```
C:\WIN2K\System32\cmd.exe
ΕΠΕΛΕΞΕ ΤΗΝ ΗΜΕΡΑ ΤΗΣ ΕΒΔΟΜΑΔΑΣ (1-7):3

Η 3η ΗΜΕΡΑ ΤΗΣ ΕΒΔΟΜΑΔΑΣ ΕΙΝΑΙ Η ΤΕΤΑΡΤΗΠΕΜΠΤΗΠΑΡΑΣΚΕΥΗΣΑΒΒΑΤΟΚΥΡΙΑΚΗΠρέπει να δώ
σεις 1-7
Press any key to continue . . .
```