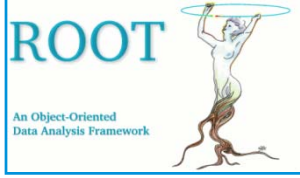
Επιλογή - Επανάληψη

- Η εντολή if-else.
- Ο τελεστής παράστασης συνθήκης
- Η εντολή switch.
- Η εντολές for και while.
- Η εντολή do-while.
- Η εντολές break - continue - goto.
- Μαθηματικές συναρτήσεις.
- Λέξεις κλειδιά στη C++.

Η εντολή if-else



- Η απλή μορφή της εντολής **if** είναι η ακόλουθη:

```
if (συνθήκη) {
        εντολή_1;
        εντολή_2;
        .
        .
}
```

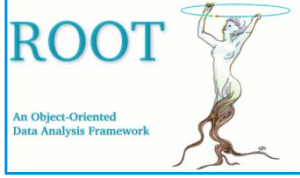
- Η μορφή της εντολής **if** με μία μόνο εντολή είναι η ακόλουθη:

```
if (συνθήκη)
        εντολή_1;
```

- Η γενική μορφή της εντολής **if-else** είναι η ακόλουθη:

```
if (συνθήκη) {
        εντολή_1;
        εντολή_2;
        .
}
else {
        εντολή_A;
        εντολή_B;
        .
}
```

Η εντολή if-else

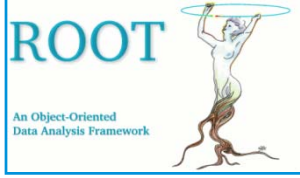


- Η γενική μορφή της εντολής **if-else** συμπεριλαμβανομένης και της **else-if** είναι:

```
if (συνθήκη_1) {
        εντολή_1_1;
        εντολή_1_2;
        .
}
else if(συνθήκη_2)  {
        εντολή_2_1;
        εντολή_2_2;
        .
}
```

```
else if(συνθήκη_3)  {
        εντολή_3_1;
        εντολή_3_2;
        .
}
else {
        εντολή_A;
        εντολή_B;
        .
}
```

Ο τελεστής παράστασης συνθήκης



- Ο τελεστής παράστασης συνθήκης (conditional expression operator) "**?:**" έχει την ακόλουθη σύνταξη:

  συνθήκη ? παράσταση_1 : παράσταση_2

  και αντιστοιχεί στην ακόλουθη σύνταξη της εντολής if:

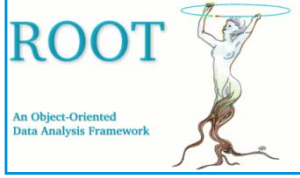
```
if(συνθήκη )
    παράσταση_1;
else
    παράσταση_2;
```

- Ο τελεστής παράστασης συνθήκης **?:** παρέχει απλά έναν εναλλακτικό τρόπο γραφής της απλής εντολής if. Στη C αναφέρεται και ως τριαδικός τελεστής.
- Για παράδειγμα, η επόμενη ανάθεση τιμής

```
min=(x<y ? x : y);
```

  θα εκχωρήσει στη min το μικρότερο από τις x και y.

Η εντολή switch.

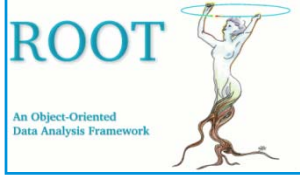


- Η γενική μορφή της εντολής **switch** είναι η ακόλουθη:

```
switch (παράσταση) {
        case σταθερή-παράσταση_1 :
                εντολή_1_1;
                εντολή_1_2;
                .
                break;
        case σταθερή-παράσταση_2 :
                εντολή_2_1;
                εντολή_2_2;
                .
                break;
        .
        .
                .
                .
                default :
                        εντολή_A;
                        εντολή_B;
                        .
                        break;
    }
```

- Πρώτα υπολογίζεται η ***παράσταση*** και στη συνέχεια εκτελείται η περίπτωση (**case**) της οποίας η ***σταθερή-παράσταση*** ταυτίζεται με αυτή.

Παράδειγμα με την εντολή if

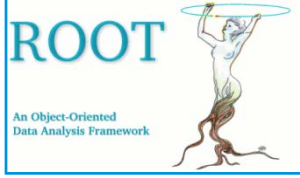


- Να γραφεί ένα πρόγραμμα το οποίο να συγκρίνει μεταξύ τους τρεις ακεραίους αριθμούς και να υπολογίζει τον μικρότερό τους. Οι τρεις ακέραιοι να εισαχθούν στο πρόγραμμα από το πληκτρολόγιο. Να χρησιμοποιήσετε την εντολή **if**.
- Ακολουθούν τρεις δυνατές λύσεις:

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

int main()
    { // finds the minimum of three input integers:
    int n1, n2, n3;
    cout << "Enter three integers: ";
    cin >> n1 >> n2 >> n3;
    int min=n1;                 // now min <= n1
    if (n2 < min) min = n2;     // now min <= n1 and min <= n2
    if (n3 < min) min = n3;     // now min <= n1, min <= n2, and min <= n3
    cout << "Their minimum is " << min << endl;
}
```

Παράδειγμα με την εντολή if

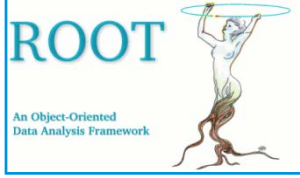


- Όπως και στην προηγούμενη λύση έτσι και στην παρακάτω χρήση της απλής εντολής if.

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

int main()
{ // finds the minimum of three input integers:
     int n1, n2, n3;
     cout << "Enter three integers: ";
     cin >> n1 >> n2 >> n3;
     if (n1 <= n2 && n1 <= n3) cout << "Their minimum is " << n1 <<endl;
     if (n2 <= n1 && n2 <= n3) cout << "Their minimum is " << n2 <<endl;
     if (n3 <= n1 && n3 <= n2) cout << "Their minimum is " << n3 <<endl;
}
```

Παράδειγμα με την εντολή if

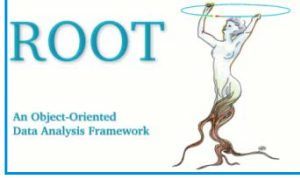


- Στη παρακάτω λύση γίνεται χρήση της εντολής if-else.

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

int main()
{ // finds the minimum of three input integers:
     int n1, n2, n3;
     cout << "Enter three integers: ";
     cin >> n1 >> n2 >> n3;
     if (n1 < n2)
        if (n1 < n3) {cout << "Their minimum is " << n1 << endl;}
        else {cout << "Their minimum is " << n3 << endl;}
     else   // n1 >= n2
        if (n2 < n3) {cout << "Their minimum is " << n2 << endl;}
        else  {cout << "Their minimum is " << n3 << endl;}
}
```

Η εντολές for και while



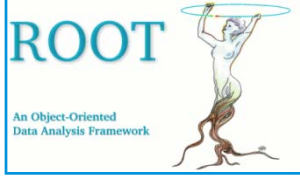
- Η γενική σύνταξη της εντολής **for** είναι η ακόλουθη:

```
for (απόδοση_αρχικής_τιμής; συνθήκη; ενημέρωση){
        εντολή_1;
        εντολή_2;
        .
        .
}
```

- Η γενική σύνταξη της εντολής **while** είναι η ακόλουθη:

```
while(συνθήκη){
        εντολή_1;
        εντολή_2;
        .
        .
}
```

Η εντολές for και while



- Η εντολή **for** μπορεί να γραφεί χρησιμοποιώντας την εντολή **while** ως:

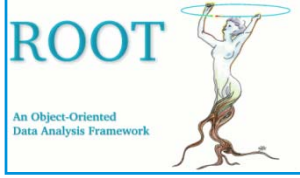
```
απόδοση_αρχικής_τιμής ;
while (συνθήκη){
        εντολή_1;
        εντολή_2;
        .
        ενημέρωση ;
}
```

- Η δημιουργία ενός ατέρμονα βρόγχου γίνεται ως εξής:

```
for (;;){
    εντολή_1;
    εντολή_2;
    .
}
```

```
while(true){
    εντολή_1;
    εντολή_2;
    .
}
```

Η εντολή do-while

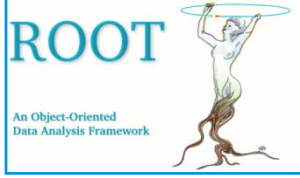


- Η γενική σύνταξη της εντολής **do-while** είναι η ακόλουθη:

```
do{
    εντολή_1;
    εντολή_2;
    .
    .
} while(παράσταση);
```

- Η εντολή **do-while**, όπως και οι εντολές for και while, χρησιμοποιείται για τη δημιουργία βρόγχων.
- Ο έλεγχος στην εντολή **do-while** γίνεται στο τέλος του βρόγχου σε αντίθεση με τις εντολές for και while στις οποίες ο έλεγχος γίνεται στην αρχή του βρόγχου. Αυτό σημαίνει πως το σώμα της εντολής εκτελείται τουλάχιστον μία φορά.

Παράδειγμα με την εντολή for

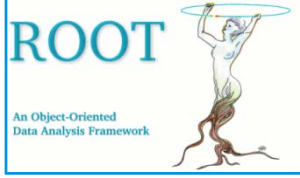


- Να γράψετε ένα πρόγραμμα το οποίο να τυπώνει δέκα ψευδοτυχαίους ακεραίους αριθμούς. Στο αρχείο-κεφαλίδα <cstdlib> ορίζονται τα ακόλουθα:
  - RAND_MAX : ο μέγιστος ακέραιος
  - int rand(): η συνάρτηση επιστρέφει έναν τυχαίο αριθμό από 0 έως RAND_MAX.
  - void srand(unsigned int seed) : η συνάρτηση αυτή εκτελείται μια φορά στην αρχή του προγράμματος και αρχικοποιεί την γεννήτρια των τυχαίων αριθμών σύμφωνα με τον ακέραιο seed

```
//  Παραγωγή 10 ψευδοτυχαίων ακεραίων αριθμών
#include <cstdlib>   // Ορίζει τα RAND_MAX , rand() και srand()
#include <iostream>
using namespace std;

int main()
{ // τυπώνει 10 ψευδοτυχαίους αριθμούς:
      srand(1234);    // αρχικοποίηση γεννήτριας με ακέραιο της αρεσκείας μας
      for (int i = 0; i < 10; i++){
           cout << rand() << endl;
       }
      cout << "RAND_MAX = " << RAND_MAX << endl;
}
```

Παράδειγμα στην εντολή while

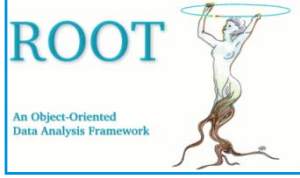


- Να γράψετε ένα πρόγραμμα το οποίο να τυπώνει δέκα ψευδοτυχαίους αριθμούς κινητής υποδιαστολής από το 20 έως το 80.

```
//  Παραγωγή 10 ψευδοτυχαίων αριθμών κινητής υποδιαστολής μεταξύ 20 και 80
#include <cstdlib>   // Ορίζει τα RAND_MAX , rand() και srand()
#include <iostream>
using namespace std;

int main()
{ // τυπώνει 10 ψευδοτυχαίους αριθμούς κινητής υποδιαστολής :
      srand(8765);    // αρχικοποίηση γεννήτριας με ακέραιο της αρεσκείας μας
      int i=0;
      while(i<10) {
          double x = (double) rand()/(double)RAND_MAX;   // ψευδοτυχαίος μεταξύ 0-1
          cout << x*(80-20)+20 << endl;
          ++i;
       }
}
```

Παράδειγμα στην εντολή while

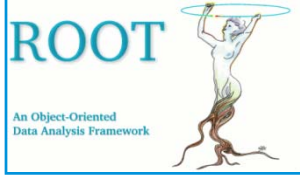


- Να γράψετε ένα πρόγραμμα το οποίο να τυπώνει δέκα ψευδοτυχαίους ακεραίους αριθμούς σε ένα διάστημα της αρεσκείας σας. Αρχικοποιείστε την γεννήτρια αριθμών με το ρολόι του συστήματος.

```
//  Παραγωγή ψευδοτυχαίων αριθμών σε συγκεκριμένο διάστημα
#include <cstdlib>
#include <ctime>      // defines the time() function
#include <iostream>
//#include <time.h>   // use this if <ctime> is not recognized
using namespace std;

int main()
{ // prints pseudo-random numbers:
      unsigned seed = time(NULL);         // uses the system clock
      cout << "seed = " << seed << endl;
      srand(seed);                        // initializes the seed
      int min, max;
      cout << "Enter minimum and maximum: ";
      cin >> min >> max;                  // lowest and highest numbers
      int range = max - min + 1;          // number of numbers in range
      for (int i = 0; i < 20; i++) {
           int r = rand()/100%range + min;
           cout << r << endl;
      }
}
```

Η εντολές break - continue - goto



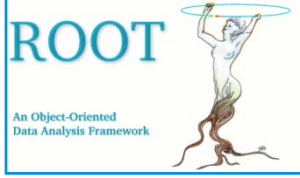
- Η εντολή **break** μας επιτρέπει να βγούμε από ένα βρόγχο χωρίς να εκτελέσουμε τον έλεγχο της αρχής ή του τέλους.

```
for( int i=1; i<=10; ++i){
    if(i==6){           // Μόλις το i πάρει την τιμή 6 εκτελείται το break
        break;
    }
    cout << i << endl;          // Εκτύπωση του i
}
```

- Η εντολή **continue** χρησιμοποιείται σε περιπτώσεις στις οποίες επιθυμούμε να παραμείνουμε σε ένα βρόγχο, αλλά να αγνοήσουμε κάποιες από τις εντολές του.

```
for(int i=1; i<=10; ++i){
    if((i==4)||(i==6)){     // Μόλις το i πάρει τις τιμή 4 ή 6 εκτελείται το continue
        continue;
    }
    cout << i << endl;          // Εκτύπωση του i
}
```

Η εντολές break - continue - goto



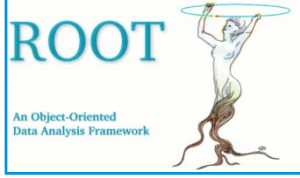
- Η εντολή **goto** χρησιμοποιείται για να κάνουμε ένα άλμα σε ένα σημείο του προγράμματος το οποίο καθορίζεται από μία ετικέτα.

```
//  Χρήση της εντολής goto για έξοδο από ένθετους βρόχους

#include <iostream>  // defines cout
using namespace std;

int main()
{ const int N=5;
     for (int i=0; i < N; i++){
          for (int j=0; j < N; j++){
               for (int k=0; k < N; k++)
                    if (i+j+k > N) goto esc;
                    else cout << i+j+k << " ";
                    cout << "* ";
          }
          esc: cout << "." << endl;  // inside the i loop, outside the j loop
    }
}
```

Μαθηματικές συναρτήσεις

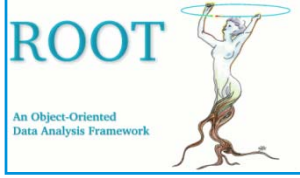


- Οι ορισμοί των βασικών μαθηματικών συναρτήσεων βρίσκονται στο αρχείο-κεφαλίδα <cmath> το οποίο πρέπει να περιλαμβάνεται στα προγράμματα μας.

Μερικές από τις συναρτήσεις που ορίζονται στην κεφαλίδα <cmath>

Συνάρτηση	Περιγραφή	Παράδειγμα
`acos(x)`	αντίστροφο συνημίτονο του x (σε ακτίνια)	Η `acos(0.2)` επιστρέφει 1.36944
`asin(x)`	αντίστροφο ημίτονο του x (σε ακτίνια)	Η `asin(0.2)` επιστρέφει 0.201358
`atan(x)`	αντίστροφη εφαπτομένη του x (σε ακτίνια)	Η `atan(0.2)` επιστρέφει 0.197396
`ceil(x)`	στρογγυλοποίηση του x στο μεγαλύτερο ακέραιο	Η `ceil(3.141593)` επιστρέφει 4.0
`cos(x)`	συνημίτονο του x (σε ακτίνια)	Η `cos(2)` επιστρέφει –0.416147
`exp(x)`	εκθέτης του x (με βάση το e)	Η `exp(2)` επιστρέφει 7.38906
`fabs(x)`	απόλυτη τιμή του x	Η `fabs(-2)` επιστρέφει 2.0
`floor(x)`	στρογγυλοποίηση του x στο μικρότερο ακέραιο	Η `floor(3.141593)` επιστρέφει 3.0
`log(x)`	φυσικός λογάριθμος του x (με βάση το e)	Η `log(2)` επιστρέφει 0.693147
`log10(x)`	κοινός λογάριθμος του x (με βάση το 10)	Η `log10(2)` επιστρέφει 0.30103
`pow(x,p)`	το x υψωμένο στη δύναμη p	Η `pow(2,3)` επιστρέφει 8.0
`sin(x)`	ημίτονο του x (σε ακτίνια)	Η `sin(2)` επιστρέφει 0.909297
`sqrt(x)`	τετραγωνική ρίζα του x	Η `sqrt(2)` επιστρέφει 1.41421
`tan(x)`	εφαπτομένη του x (σε ακτίνια)	Η `tan(2)` επιστρέφει –2.18504

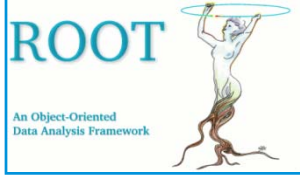
Λέξεις κλειδιά στη C++



- Λέξη κλειδί (keyword) σε μια γλώσσα προγραμματισμού είναι μια λέξη η οποία έχει ήδη οριστεί και είναι δεσμευμένη για συγκεκριμένο σκοπό στα προγράμματα αυτής της γλώσσας. Η Καθιερωμένη C++ έχει τώρα 74 λέξεις κλειδιά.

and	and_eq	asm	auto	bitand
bitor	bool	break	case	catch
char	class	compl	const	const_cast
continue	default	delete	do	double
dynamic_cast	else	enum	explicit	export
extern	dfalse	float	for	friend
goto	if	inline	int	long
mutable	namespace	new	not	not_eq
operator	or	or_eq	private	protected
public	register	reinterpret_cast	return	short
signed	sizeof	static	static_cast	struct
switch	template	this	throw	true
try	typedef	typeid	typename	using
union	unsigned	virtual	void	volatile
wchar_t	while	xor	xor_eq	

Άσκηση: Το παραγοντικό n!



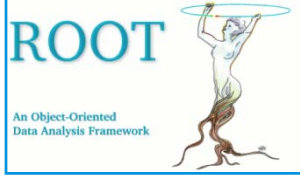
- Να γράψετε ένα πρόγραμμα το οποίο να υπολογίζει το παραγοντικό ενός αριθμού ο οποίος να εισάγεται από το πληκτρολόγιο.

```
//  Υπολογισμός n!
#include <iostream>
using namespace std;

int main()
{ // τυπώνει το n!:
      int n;
      cout << "Dose to n: " << endl;
      cin >> n;       //Εισαγωγή του n
      if(n<0) {
          cout << "n < 0 !!!!!" << endl;
          return 0;     // Τερματισμός εάν n<0
       }
      double factorial = 1;
      for (int i = 1; i <= n; i++)
          factorial *= i;

      cout << n << "! =" << factorial << endl;
}
```

Άσκηση: Οι αριθμοί Fibonacci



- Να γράψετε ένα πρόγραμμα το οποίο να τυπώνει όλους τους αριθμούς Fibonacci μέχρι το όριο που δίνει ο χρήστης.
- Οι αριθμοί Fibonacci F_0, F_1, F_2, F_3 … δίνονται αναδρομικά από τις εξισώσεις:

  $F_0=0$ $F_1=1$ $F_n=F_{n-1}+F_{n-2}$

```
//  Οι αριθμοί Fibonacci
#include <iostream>
using namespace std;

int main()
{ // prints Fibonacci numbers:
  long bound;
  cout << "Enter a positive integer: ";
  cin >> bound;
  cout << "Fibonacci numbers < " << bound << ":\n0, 1";
  long f0=0, f1=1;
  while (true) {
        long f2 = f0 + f1;
        if (f2 > bound) break;      // terminates the loop immediately
        cout << ", " << f2;
        f0 = f1;
        f1 = f2;
  }
}
```