

La programmazione *Object Oriented* e le sue applicazioni in Fisica.

Gabriella Cataldi, INFN Lecce

Daniele Martello, dip. Fisica e INFN Lecce

Un po' di sintassi

- Sintassi Fortran e C/C++ a confronto

In C/C++ non e' necessario un particolare formato di codice

```
PROGRAM TEST
C  esempio di programma
.....
spazi...
.....
END
```

Il C/C++ e' *case sensitive*

```
INTEGER I
integer j
integer*4 m,n
real x
REAL*8 D
```

```
int main() {
// esempio di programma
.....
return 0; // fine
}
```

//questo e' un commento

/* anche questo e' un commento
e puo' essere scritto su piu' linee */

istruzioni separate da ;

Input/Output

In C++ L'I/O si esegue utilizzando *oggetti* appartenenti a *classi* che rappresentano files o terminali fisici (tastiera e schermo)

OGGETTO rappresentante lo stdin

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

...

```
char ch;
```

```
cin >> ch;
```

```
cout << "Hai premuto il tasto "
      << ch << endl;
```

Operatore di estrazione

OGGETTO
rappresentante lo stdout

Operatore di inserimento

Hello, world!

- Esempio di programma semplice con I/O

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```

direttiva al preprocessore

end of line

Come compilare:

```
g++ Esercizio0.cxx -o prova
```

Tipi predefiniti in C++

	16 bit	32 bit	64 bit
char	8	8	8
int	16	32	32
bool	16	32	32
short	16	16	16
long	32	32	64
float	32	32	32
double	64	64	64
long double	128	128	128

puo' essere unsigned

Operazioni e Operatori

```
int c=0;  
float d;  
d=3.5;  
double t(4.56);
```

L'operatore di assegnazione (=) può essere utilizzato:

- Contestualmente alla dichiarazione
- Differito rispetto a questa

E' possibile assegnare un valore attraverso il *costruttore*.

Oltre alle normali operazioni aritmetiche e' possibile usare degli operatori speciali:

- Incremento, decremento, auto-moltiplicazione e divisione
- Auto pre(post)-incremento(decremento)

```
c=c+3;    c+=3;    c/=2;  
d+=1;    d++;    ++d;
```

Operazioni e Operatori

$x+y$
 $x-y$
 $x*y$
 x/y

Le quattro operazioni base

$x=y$
 $x+=y$
 $x-=y$
 $x*=y$
 $x/=y$

Operatori di assegnazione.
Sono operatori che permettono di assegnare il valore di una variabile ad un'altra o di alterarne il valore.

$x++$
 $++x$
 $x--$
 $--x$

Incrementi e decrementi.
Questi operatori permettono di incrementare o decrementare il contenuto di una variabile.

$x+=y \rightarrow x = x + y;$

$x-=y \rightarrow x = x - y;$

$x*=y \rightarrow x = x * y;$

$x/=y \rightarrow x = x / y;$

$z = ++x \rightarrow$ $x = x + 1;$
 $z = x;$

$z = --x \rightarrow$ $x = x - 1;$
 $z = x;$

$z = x++ \rightarrow$ $z = x;$
 $x = x + 1;$

$z = x-- \rightarrow$ $z = x;$
 $x = x - 1;$

Esercizio.....

Cicli

```
while (condizione) {  
    ...  
}
```

```
int c = 0;  
while (c<10) {  
    cout << c++ <<  
endl;  
}
```

```
do {  
    ...  
} while(condizione);
```

```
int c = 0;  
do {  
    cout << c++ << endl  
} while(c<10);
```

```
for (istr_1; condizione; istr_n) {  
    cout << c << endl;  
}
```

```
for (int c=0; c<10; c++) {  
    cout << c << endl;  
}
```

Controllo di flusso

```
if (condizione) { ... };
```

```
if (a!=0) { b /= a; };
```

```
if (condizione) {  
    ...  
} else {  
    ...  
};
```

Gli operatori logici sono:

- == (uguaglianza)
- != (disuguaglianza)
- < o <= (minorazione)
- > o >= (maggiorazione)
- && (AND)
- || (OR)
- ! (NOT)

```
if (a>0) {  
    b = a;  
} else {  
    b = -a;  
};
```

Funzioni matematiche

- In C++ non esistono funzioni predefinite

```
#include <iostream>
#include <cmath>
int main(){
    double r, theta, phi;
    cin >>r>>theta>>phi;
    double x = r * sin(theta) * sin(phi);
    double y = r * sin(theta) * cos(phi);
    double z = r *cos(theta);

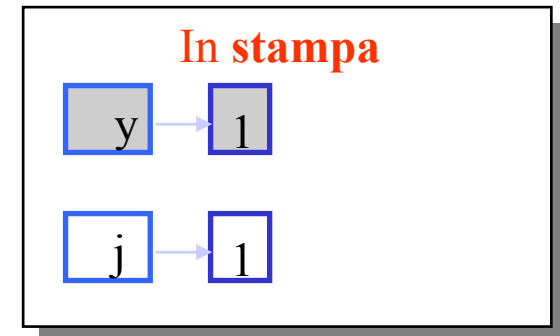
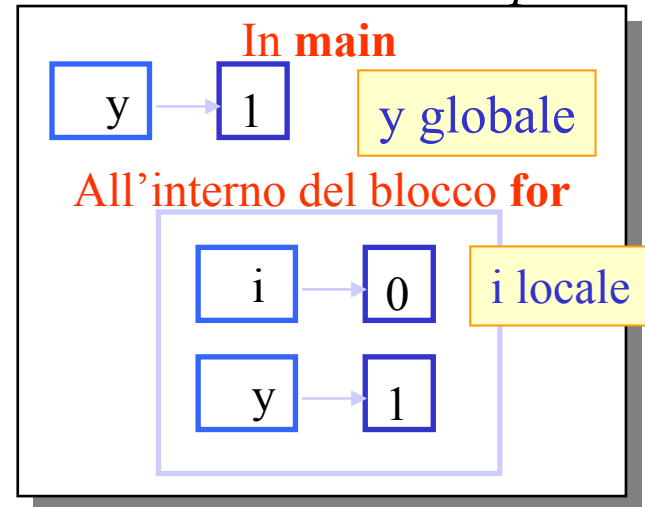
    cout <<x <<" "<<y<<" "<<z<<endl;
    return 0;
}
```

cmath.h definisce sin, cos, ...

Visibilit  delle variabili

- Le dichiarazioni possono essere fatte ovunque nel codice.
- Le dichiarazioni di variabili sono valide solo entro un certo *scope*.

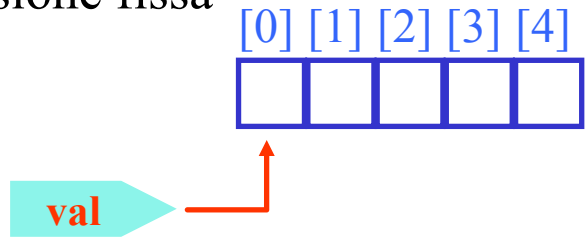
```
#include <iostream>
void stampa();
void main() {
    double y=1.;
    for (int ind=0; ind<3; ind++) {
        int i=0;
        cout <<"i="<< i <<"", y="<<y<<endl;
    }
    cout << "i="<<i<<endl; /* errore! non compila */
    stampa();
}
void stampa() {
    int j=1;
    cout <<"y="<< y <<"", j="<<j<<endl;
}
```



Array

- In C++ sono supportati gli *array* di dimensione fissa

```
#include <iostream>
/* Ricerca del massimo in un set di valori */
void main () {
    int num_val, ind=0;
    float maxval=0., val[5];
    cout<<"inserisci il numero di valori "<< endl;
    cin>>num_val;
    /* Loop principale */
    while(ind<num_val){
        cout<<"inserisci i valori "<< endl;
        cin>>val[ind];
        if (ind==0) maxval=val[ind];
        if (val[ind]>maxval) maxval=val[ind];
        ind++;
    }
    cout<<"Il massimo valore inserito e' "<<
    cout <<maxval<<endl;
}
```



L'indice va da 0 a 4
*Usare un indice maggiore di 4
puo' causare un crash*

```
// -----  
// Questo e' un commento!  
//  
// Programma modificato da:  
//  
// -----  
#include <iostream> // direttiva al preprocessore  
#include "Statistica.h"  
using namespace std;  
int main()  
{  
    /* Anche questi sono commenti!  
       cout stampa(=out) operatore di inserimento <<  
       cin  legge(=in) operatore di estrazione >>  
    */
```

//definizione e inizializzazione di variabili

double x[5]={10.3,10.5,12.4,13.5,12.6};

double dx[5]={0.1,0.2,0.3,0.1,0.2};

int N=5;

//

cout << "Media Aritmetica = "

<< MediaAritmetica(x,dx,N)<<endl;

cout << "Media Geometrica = "

<< MediaGeometrica(x,dx,N)<<endl;

cout << "Media Pesata = "

<< MediaPesata(x,dx,N)<<endl;

cout << "Deviazione Standard = "

<< DeviazioneStandardDati(x,dx,N)<<endl;

```

cout << "Errore Media Aritmetica  = "
    << ErroreMediaAritmetica(x,dx,N)<<endl;
cout << "Errore Media Pesata      = "
    << ErroreMediaPesata(x,dx,N)<<endl;
cout << "*****" << endl;
cout << " Funzione di Probabilita' Gaussiana " << endl;
cout << "*****" << endl;
double Media=MediaAritmetica(x,dx,N);
double Sigma=DeviazioneStandardDati(x,dx,N);
for (int i=0; i < N ; i++ ){
    cout << FunzioneProbGaussiana(x[i],Media,Sigma)<<endl;
}
cout << "*****" << endl;
cout << " Integrale di Funzione di Probabilita' Gaussiana " << endl;
cout << "*****" << endl;
for (int i=0; i < N ; i++ ){
    cout << IntegraleProbGauss(x[i],Media,Sigma)<<endl;
}
return 0;
}

```

Programmi piu' complessi

- Se si hanno piu' files da compilare e *linkare*:
 - compilazione senza load(link) dei singoli files:

```
g++ -c a.cxx -o a.o
```

```
g++ -c b.cxx -o b.o
```

- load(link) di tutti i files:

```
g++ Main.cxx a.o -o mytest
```

```
g++ Main.cxx *.o -o mytest
```