

# Breve Manuale di Riferimento sulla Sintassi Linguaggi C++ e FORTRAN

versione aggiornata al

05/02/2004

## Sommario

|   |          |
|---|----------|
| <b>Elementi di Base .....</b>                         | <b>2</b> |
| <b>Dati, Tipi e Dichiarazioni .....</b>               | <b>2</b> |
| <i>Tipi di Dato di Base .....</i>                     | <i>2</i> |
| <i>Tipi di Dato Strutturati.....</i>                  | <i>2</i> |
| <i>Dichiarazioni di Variabili .....</i>               | <i>2</i> |
| <i>Dichiarazioni di Costanti .....</i>                | <i>3</i> |
| <b>Espressioni e Assegnazioni .....</b>               | <b>3</b> |
| <i>Operatori e Funzioni Predefinite.....</i>          | <i>3</i> |
| <i>Istruzioni di Assegnazione.....</i>                | <i>3</i> |
| <b>Lettura e Stampa .....</b>                         | <b>3</b> |
| <i>Procedure Predefinite di Lettura e Stampa.....</i> | <i>3</i> |
| <b>Strutture di Controllo.....</b>                    | <b>4</b> |
| <i>Istruzioni Condizionali.....</i>                   | <i>4</i> |
| <i>Cicli di tipo "FOR" .....</i>                      | <i>4</i> |
| <i>Cicli di tipo "WHILE" .....</i>                    | <i>4</i> |
| <b>Sottoprogrammi.....</b>                            | <b>5</b> |
| <i>Procedure .....</i>                                | <i>5</i> |
| <i>Funzioni.....</i>                                  | <i>5</i> |
| <b>Principali Differenze tra C++ e C .....</b>        | <b>6</b> |
| <i>Costanti .....</i>                                 | <i>6</i> |
| <i>Tipi di Dato.....</i>                              | <i>6</i> |
| <i>Lettura e Stampa.....</i>                          | <i>6</i> |
| <i>Sottoprogrammi .....</i>                           | <i>6</i> |

### **Nota:**

Questo manuale è finalizzato esclusivamente allo svolgimento delle prove del corso di Informatica della Facoltà di Ingegneria dell'Università della Basilicata. I contenuti del manuale rispecchiano gli argomenti trattati nel corso e non devono essere intesi in alcun modo come un riferimento esaustivo sulla sintassi dei linguaggi trattati.

| <b>Elementi di Base</b>   |   |
|---|---|
| C++   | FORTRAN   |
| <p>Struttura del programma:</p> <pre>void main(){   &lt;corpo del programma&gt; }</pre> <p>Parole chiave:</p> <pre>void main</pre> <p>Commenti:</p> <pre>// &lt;testo del commento&gt; (una riga) /* &lt;testo del commento&gt; (anche su più   righe) */</pre> | <p>Struttura del programma:</p> <pre>PROGRAM &lt;identificatore&gt; IMPLICIT NONE   &lt;corpo del programma&gt; STOP END</pre> <p>Parole chiave:</p> <pre>PROGRAM END STOP IMPLICIT NONE</pre> <p>Commenti:</p> <pre>C &lt;testo del commento&gt;</pre> |

| <b>Dati, Tipi e Dichiarazioni</b>  |  |
|--|--|
| Tipi di Dato di Base   |  |
| C++  | FORTRAN  |
| <p>Parole chiave:</p> <pre>int float double bool char string</pre>   | <p>Parole chiave:</p> <pre>INTEGER REAL DOUBLE PRECISION LOGICAL CHARACTER CHARACTER *n COMPLEX</pre>  |
| Tipi di Dato Strutturati   |  |
| C++  | FORTRAN  |
| <p>Parole chiave</p> <pre>struct</pre> <p>(in C++ sono disponibili ARRAY mono e pluridimensionali e RECORD, detti STRUTTURE)</p> <p>Esempio:</p> <p>vedi sotto</p>   | <p>Parole chiave:</p> <pre>&lt;nessuna&gt;</pre> <p>(in FORTRAN sono disponibili ARRAY mono e pluridimensionali, ma non sono disponibili i RECORD)</p> <p>Esempio:</p> <p>vedi sotto</p> |
| Dichiarazioni di Variabili   |  |
| C++  | FORTRAN  |
| <p>Parole chiave:</p> <pre>&lt;nessuna&gt;</pre> <p>Esempi:</p> <pre>int x, y; string cognome; bool trovato; int v[10]; float m[5][5]; struct studente {   string nome;   int annoCorso; }; studente stud;</pre> | <p>Parole chiave:</p> <pre>&lt;nessuna&gt;</pre> <p>Esempi:</p> <pre>INTEGER x, y CHARACTER*20 cognome LOGICAL trovato INTEGER v(10) INTEGER s (-1:100) REAL m(5,5)</pre>                |

| <b>Dichiarazioni di Costanti</b>  |  |
|---|--|
| C++   | FORTRAN  |
| Parole chiave:<br>const<br><br>Esempio:<br>const int N=10;<br>const float pigreco = 3.14;<br>const char segno = 'X';<br>const string corso = "Informatica"; | Parole chiave:<br>PARAMETER<br><br>Esempio:<br>INTEGER N<br>PARAMETER (N = 10)<br>REAL pigreco<br>PARAMETER (pigreco = 3.14)<br>CHARACTER segno<br>PARAMETER (segno = 'X') |

| <b>Espressioni e Assegnazioni</b>   |  |
|---|--|
| <b>Operatori e Funzioni Predefinite</b>   |  |
| C++   | FORTRAN  |
| Principali operatori:<br>+ - * / %<br>&&    !<br>== > >= < <= !=<br><br>Principali funzioni predefinite:<br>abs() fabs() pow() sqrt() exp() cos()<br>atan() log() log10() floor() | Principali operatori:<br>+ - * / **<br>.AND. .OR. .NOT. .EQV. .NEQV.<br>.EQ. .GT. .GE. .LT. .LE. .NE.<br><br>Principali funzioni predefinite:<br>ABS() MOD() SQRT() EXP() LOG() SIN()<br>COS() TAN() ATAN() AINT() ANINT() |

| <b>Istruzioni di Assegnazione</b>   |  |
|---|--|
| C++   | FORTRAN  |
| Parole chiave:<br>=<br><br>Esempio:<br>int x,y;<br>bool z;<br>x = 3;<br>x = x + 1;<br>x = pow(y, 2) + 3 - floor(5.20);<br>z = (x > y) && (y <= 10); | Parole chiave:<br>=<br><br>Esempio:<br>INTEGER x,y<br>LOGICAL z<br>x = 3<br>x = x + 1<br>x = sqr(y) + 3 - aint(5.20)<br>z = (x.GT.y).AND.(y.LT.10) |

| <b>Lettura e Stampa</b>  |  |
|--|--|
| <b>Procedure Predefinite di Lettura e Stampa</b>   |  |
| C++  | FORTRAN  |
| Lettura da tastiera:<br>>> operatore di lettura<br>cin input stream<br><br>Stampa su video:<br><< operatore di stampa<br>cout output stream<br><br>Esempio:<br>float x;<br>cout << "Immetti il valore di x: ";<br>cin >> x;<br>cout << "Il valore di x e' :" << x; | Procedure di lettura da tastiera:<br>READ(*,*)<br><br>Procedure di stampa su video:<br>WRITE(*,*)<br>WRITE(*, <etichetta formato>)<br><br>Esempio:<br>REAL x<br>write(*,*) 'Immetti il valore di x:'<br>read (*,*) x<br>write(*,100) 'Il valore di x e'' :', x<br>100 format (A20, F6.2) |

| <b>Strutture di Controllo</b>  |   |
|--|---|
| <b>Istruzioni Condizionali</b>   |   |
| C++  | FORTRAN   |
| <p>Parole chiave:<br/>if else</p> <p>Esempio:<br/>int x,y;</p> <pre>if (x&gt;2) cout &lt;&lt; x;</pre> <pre>if ((x==3)    (x==5)) {     cout &lt;&lt; x;     x = x+1; };</pre> <pre>if (x&gt;0) {     cout &lt;&lt; "X e' positivo"; } else {     if (x==0) {         cout &lt;&lt; "X e' nullo";         y = x + 1;     } else {         cout &lt;&lt; "X e' negativo";         y = -x;     } }</pre> | <p>Parole chiave:<br/>IF THEN ELSE ENDIF</p> <p>Esempio:<br/>INTEGER x,y</p> <pre>IF (x.GT.2) THEN     write(*,*) x ENDIF</pre> <pre>IF ((x.EQ.3).OR.(x.EQ.5)) THEN     write (*,*) x     x = x+1 ENDIF</pre> <pre>IF (x.GT.0) THEN     write (*,*) 'X e'' positivo' ELSE     IF (x.EQ.0) THEN         write (*,*) 'X e'' nullo'         y = x+1     ELSE         write (*,*) 'X e'' negativo'         y = -x     ENDIF ENDIF</pre> |
| <b>Cicli di tipo "FOR"</b>   |   |
| C++  | FORTRAN   |
| <p>Parole chiave:<br/>for</p> <p>Esempio:<br/>int i,j;</p> <pre>for (i = 1; i &lt;= 10; i++) {     cout &lt;&lt; i &lt;&lt; endl; }</pre> <pre>for (i = 1; i &lt;= 10; i++) {     for (j = 1; j &lt;= 10; j++) {         cout &lt;&lt; i*j;     }     cout &lt;&lt; endl; }</pre>  | <p>Parole chiave:<br/>DO<br/>END DO</p> <p>(in FORTRAN è consentito anche l'utilizzo di cicli DO impliciti)</p> <p>Esempio:<br/>INTEGER i,j</p> <pre>DO 100 i=1,10     write (*,*) i END DO</pre> <pre>DO 200 i=1,10     DO 250 j=1,10         write(*,*) i*j     END DO END DO</pre> <pre>DO 300 i=1,10     write(*,*) (i*j, j=1,10) C    utilizzo del do implicito END DO</pre>   |
| <b>Cicli di tipo "WHILE"</b>   |   |
| C++  | FORTRAN   |
| <p>Parole chiave:<br/>while</p> <p>Esempio:<br/>int i,j;</p> <pre>i=1;</pre>   | <p>Parole chiave:<br/>WHILE<br/>DO<br/>ENDDO</p> <p>Esempio:<br/>INTEGER i,j</p>  |

|   |  |
|---|--|
| <pre>while (i &lt;= 10) {     cout &lt;&lt; i &lt;&lt; endl;     i++; }  i = 1; while(i &lt;= 10) {     j = 1;     while(j &lt;= 10) {         cout &lt;&lt; i*j;         j = j + 1;     }     cout &lt;&lt; endl;     i++; }</pre> | <pre>i=1 DO WHILE (i.LE.10)     write (*,*) i     i=i+1 END DO  i=1 DO WHILE (i.LE.10)     j=1     DO WHILE (j.LE.10)         write (*,*) i*j         j=j+1     ENDDO     i=i+1 END DO  i=1 DO WHILE (i.LE.10)     write(*,*) (i*j, j=1,10)     utilizzo del do implicito     i=i+1 END DO</pre> |
|---|--|

| <b>Sottoprogrammi</b>   |  |
|---|--|
| Procedure   |  |
| C++   | FORTRAN  |
| <p>Parole chiave:<br/>return</p> <p>Esempio:</p> <pre>void scambia (int&amp; x, int&amp; y){     int c;     c = x;     x = y;     y = c;     return; }  void main() {     int a, b;     a = 1;     b = 2;     cout &lt;&lt; a &lt;&lt; b &lt;&lt; endl;     scambia (a, b);     cout &lt;&lt; a &lt;&lt; b &lt;&lt; endl; }</pre> | <p>Parole chiave:<br/>SUBROUTINE<br/>RETURN<br/>CALL</p> <p>Esempio:</p> <pre>SUBROUTINE scambia (x, y)     INTEGER x,y     INTEGER c     c = x     x = y     y = c     RETURN END  PROGRAM prova     INTEGER a,b     a = 1     b = 1     write (*,*) a, b     CALL scambia (a, b)     write (*,*) a, b     STOP END</pre> |
| Funzioni  |  |
| C++   | FORTRAN  |
| <p>Parole chiave:<br/>return &lt;identificatore&gt;</p> <p>Esempio:</p> <pre>float modulo (int x, int y){     float c;     c = sqrt(pow(x,2)+pow(y,2));     return c; }  void main() {     int a, b;     a = 1;     b = 2;     cout &lt;&lt; modulo(a,b) &lt;&lt; endl; }</pre>   | <p>Parole chiave:<br/>FUNCTION<br/>RETURN</p> <p>Esempio:</p> <pre>REAL FUNCTION modulo (x, y)     INTEGER x,y     REAL c     c = sqrt(x * x + y * y)     modulo = c     RETURN END  PROGRAM prova     INTEGER a,b</pre>   |

|  |   |
|--|---|
|  | REAL modulo<br>C dichiarazione della funzione utilizzata<br>a = 1<br>b = 1<br>write (*,*) modulo(a, b)<br>STOP<br>END |
|--|---|

## Principali Differenze tra C++ e C

### Costanti

In C le costanti si definiscono utilizzando la direttiva #define; Esempio:

```
#define PIGRECO 3.14
```

### Tipi di Dato

Il C non prevede il tipo bool. Normalmente viene surrogato utilizzando una variabile di tipo int a cui si attribuiscono valori 0 e 1.

Il C non prevede il tipo string. Normalmente viene surrogato utilizzando un array di caratteri.

La dichiarazione delle strutture in C viene fatta utilizzando una sintassi leggermente diversa. Esempio:

```
typedef struct {
    char nomer[20];
    int annoDiCorso;
} studente;
```

```
studente s;
```

### Letture e Stampa

La libreria per la gestione dei flussi standard in C si chiama stdio.h; l'istruzione per la stampa sullo standard output è printf; l'istruzione per la lettura dallo standard input è scanf. Esempi:

```
#include <stdio.h>

int main() {
    int x;
    printf("Immetti il valore di x");
    scanf ("%d", &x);
    printf("Valore di x: %d", x);
}
```

Descrittori di formato:

%d: numero intero

%f: numero reale

%c: carattere singolo

%s: sequenza di caratteri (stringa)

### Sottoprogrammi

Il C non prevede il tipo di riferimento e quindi il passaggio dei parametri per riferimento. L'effetto del passaggio per riferimento viene normalmente ottenuto utilizzando parametri di tipo puntatore.