

Chapter 0 Programming in Computer Languages

0-1 Main Programs

C++ language:

```
#include <stdio.h>
#include <math.h>
main ()
{
statement 1; statement 2; statement 3;
statement 4; ...
...
}
```

FORTRAN language: (In *filename.for*, each statement starts from the 7th column and is terminated on the 80th column. If the length of one statement beyond the 80th column and extends in the next row, the 6th column of the next row must be typed "+". But in *filename.f90*, no such constraints exist)

```
statement 1
statement 2
statement 3
...
stop
end
```

MATLAB language:

```
statement 1; statement 2;
statement 3;
statement 4; ...
```

0-2 Declarations

C++ language:

```
float n[3],m[2][3],p[4][3][4],a,b;
int c[4],m[4][3],d;
char e,f;
```

Note: $n[k]$ starts from $n[0]$ to $n[k-1]$; $m[i][j]$ starts from $m[0][0]$ to $m[i-1][j-1]$.

If one wants to declare double $e1[101][201][301]$ in C++ language:

```
#include <stdio.h>
#include <math.h>
main ()
{
int i,j,k;
const int size1=101,size2=201,size3=301;
double ***e1;
e1=new double**[size1]; // 配置第一維
for(i=0; i<size1; i++)
    {
        e1[i]=new double*[size2]; // 配置第二維
        for(j=0; j<size2; j++)
            {
                e1[i][j]=new double[size3]; // 配置第三維
                for(k=0; k<size3; k++)
                    {
                        e1[i][j][k]=0;
                    }
            }
    }
.....
for(i=0; i<size1; i++)
{
    for(j=0; j<size2; j++)
    {
        delete [] e1[i][j]; // 釋放第三維
    }
    delete [] e1[i]; // 釋放第二維
}
delete [] e1; // 釋放第一維
}
```

FORTRAN language:

```
real n(3),m(2,3),p(4,3,4),u(0:2),v(-1:1),w(-5:5,-3:2),a,b
integer c(4),mm(4,3),d
character e,f
```

Note:

1. In FORTRAN language, $n(3)$ starts from $n(1)$ to $n(3)$; $u(0:2)$ starts from $u(0)$ to $u(2)$; $v(-2:1)$ starts from $v(-2)$ to $v(1)$; $w(-5:5,-3:2)$ starts from $w(-5,-3)$ to $w(5,2)$.
2. In FORTRAN language, all the declarations must be placed on the start of the program.
3. In case of no declarations, variables with the first letter $i, j, k, l, m,$ and n will be automatically seen as integers in the FORTRAN program.
4. In FORTRAN language, $x=X, yz=YZ=yZ=Yz,$ etc. But in C++ language, $x\neq X,$ etc.

0-3 Inputs and Outputs

C++ language:

```
printf ("A+B"); printf("C+D"); // print A+B and C+D on the screen.
```

```
printf ("A+B \n"); printf("C+D \n"); // print A+B
```

$C+D$ on the screen.

```
printf ("%f %d", x, i); // print the values of x (float number) and i (integer) on the screen.
```

```
scanf ("%f %d", &x, &i); // input x (float number) and i (integer) by the keyboard.
```

FORTRAN language:

```
write (*,*) 'C+D' (print C+D on the screen)
```

```
write (*,*) a (print the value of a on the screen)
```

```
read (*,*) a (input the value of a by the keyboard)
```

MATLAB language:

```
input a; %input the value of a from keyboard
```

```
disp X; %display letter X
```

```
format short %輸出有效數字 5 位
```

```
format short e
```

```
format long %輸出有效數字 15 位
```

```
format long e
```

0-4 Arithmetic Expressions and Loops

C++ language:

```
for(int i=1;i<=10;i++)
{
    for(int j=-2;j<=5;j++)
    {
        for(int k=0;k<=3;k++)
        {
            int n=i+j+k; float p=(i*j-k)/2; float z=pow(i,2)+pow(j,3)+pow(k,4);
        }
    }
}
```

FORTRAN language: (All the labels are placed in from the second to the fifth columns)

```
do 1 i=1,10
do 2 j=-2,5
do 3 k=0,3
n=i+j+k
p=( i*j-k)/2
z=i**2+j**3+k**4
3 continue
2 continue
1 continue
```

MATLAB language:

```
for i=1:10
for j=-2:5
for k=0:3
n=i+j+k;
p=( i*j-k)/2;
z=i^2+j^3+k^4;
end
end
end
```

Note:

In C++ language, we have 2 other major types of loops such as:

1.

```
int i=1; int sum=0;
while (i<=10)
{
    sum=sum+i;
    i++;
}
```

2.

```
int i=1; int sum=0;
do {
    sum=sum+i;
    i++;
} while (i<=10);
```

In FORTRAN language, we can have the loop counter with increment other than

1:

```
      do 10 i=1,10,3      (i=1, 4, 7, 10 with increment 3)
          statement 1
          statement 2
          statement 3
          ...
10    continue
```

In MATLAB language, we have the types of loops such as:

1. `sum=0; i=0;`

`while i<10`

`sum=sum+i;`

`i=i+1;`

`end`

2. `for i=0.2:0.3:1.1` % i starts from 0.2 to 1.1 with increment 0.3

`statement 1; statement 2;`

`statement 3; ...`

`end`

Eg. Calculate π in C++ language by $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - + \dots = \sum_{m=0}^{\infty} \frac{(-1)^m}{(2m+1)} = \frac{\pi}{4}$.

Case 1 in C++ language:

```
#include <stdio.h>
#include <math.h>
main()
{
float sum=0;
int im=10000;
    for(int i=0;i<=im;i++)
    {
        sum=sum+pow(-1,i)/(2*i+1);
    }
    sum*=4;
    printf("%f",sum);
}
```



Case 2 in C++ language:

```
#include <stdio.h>
#include <math.h>
main(){
float sum=1,term,er;
int i=1;
    do
    {
        term=pow(-1,i)/(2*i+1);
        sum=sum+term;
        er=term*term;
        i++;
    } while (er>0.000001);
    sum*=4;
    printf("%f  %d",sum,i);
}
```



0-5 If Conditions

C++ language:

```
if(a>=90)
{statement 1; statement 2; statement 3; ...}
else if (a==80)
{statement n1; statement n2; statement n3; ...}
else
{statement m1; statement m2; statement m3; ...}
```

FORTRAN language:

```
if (a.ge.90) then
  statement 1
  statement 2
  ...
elseif (a.eq.80) then
  statement n1
  statement n2
  ...
else
  statement m1
  statement m2
  ...
Endif
```

MATLAB language:

```
if a>=90
  statement 1; statement 2; statement 3; ...
elseif a==80
  statement n1; statement n2; statement n3; ...
else
  statement m1; statement m2; statement m3; ...
end
```

Relation	C++	FORTRAN	MATLAB
>	>	.gt.	>
<	<	.lt.	<
\geq	>=	.ge.	>=
\leq	<=	.le.	<=
\neq	!=	.ne.	~=
=	==	.eq.	==
AND	&&	.and.	&
OR		.or.	

0-6 Subroutines and Functions

Subroutines (副程式)

C++ language:

```
#include <stdio.h>
```

```
#include <math.h>
```

```
void output ()
```

```
{
```

```
    printf ("Happy Birthday!\n");
```

```
}
```

```
main ()
```

```
{
```

```
    output ();
```

```
}
```

FORTRAN language:

```
    call output
```

```
    stop
```

```
end
```

```
subroutine output
```

```
    write (*,*) 'Happy Birthday!'
```

```
    return
```

```
end
```


Functions (函數)

C++ language:

```
#include <stdio.h>
#include <math.h>
float sum(int n)
{
    int i;
    float sum=0;
    for (i=1;i<=n;i++)
    {
        sum=sum+i;
    }
    return(sum);
}

main ()
{
    int j=5;
    float total=sum(j);
    printf("%f\n",total);
}
```

FORTRAN language:

```
external sum
i=5
total=sum(i)
write (*,*) total
stop
end

function sum(j)
sum=0.
do 1 i=1,j
sum=sum+i
1 continue
return
end
```