

12.010 Basic C: C and Fortran 77 Syntax

Declarations

C (statements end in ; all variables must be declared explicitly)	F77 (start in column 7)
<code>int a;</code>	INTEGER A
<code>float a;</code>	REAL A
<code>double a;</code>	DOUBLE PRECISION A
<code>char a;</code>	CHARACTER A
<code>short a;</code>	
<code>uint a;</code>	
<code>long int a;</code>	
<code>int a[10];</code>	INTEGER A(10)
<code>float a[10][10];</code>	REAL A(10,10)
<code>char a[10];</code>	CHARACTER A(10)
<code>char a[10];</code>	CHARACTER A(10)

Simple Loop

C (array indices range 0 to N-1)

```
int i,j;
float arr[10][10];

for (j=0;j<10;++j){
  for (i=0;i<10;++i){
    arr[j][i]=1
  }
}
```

F77 (array indices range 1 to N)

```
INTEGER I
INTEGER J
REAL ARR(10,10)

DO J=1,10
  DO I=1,10
    ARR(I,J)=1
  ENDDO
ENDDO
```

C Hello World

```
#include <stdio.h>

main() {
    printf("Hello\n");           /* printf always goes to "standard
output". */
    fprintf(stdout,"Hello\n");  /* fprintf goes to specified output stream
*/
}
```

F77 Hello World

```
PROGRAM MAIN
WRITE(6,*) 'Hello'
END
```

Formatting output

(see "man fprintf")

Integers

C	F77
<pre>main() { int i; i = 7; printf("\n"); /* New line */ printf("Leading blanks\n"); printf("%1d\n",i); printf("%2d\n",i); printf("%3d\n",i); printf("%4d\n",i); printf("%5d\n",i); printf("%6d\n",i); printf("\n"); /* New line */ printf("Leading zeros\n"); printf("%1.1d\n",i); printf("%2.2d\n",i); printf("%3.3d\n",i); printf("%4.4d\n",i);</pre>	<pre>PROGRAM MAIN INTEGER I I=7 WRITE(6,'(1X)') WRITE(6,'(A)') 'Leading blanks' WRITE(6,'(I1)') I WRITE(6,'(I2)') I WRITE(6,'(I3)') I WRITE(6,'(I4)') I WRITE(6,'(I5)') I WRITE(6,'(I6)') I WRITE(6,'(1X)') WRITE(6,'(A)') 'Leading zeros' WRITE(6,'(I1.1)') I WRITE(6,'(I2.2)') I WRITE(6,'(I3.3)') I</pre>

```

printf("%5.5d\n",i);
printf("%6.6d\n",i);
}

WRITE(6,'(I4.4)') I
WRITE(6,'(I5.5)') I
WRITE(6,'(I6.6)') I

END

```

Text

<pre> C main() { printf("A"); printf(" new"); printf(" line"); printf(" must"); printf(" be"); printf(" specified"); printf(" explicitly."); printf("\n"); } </pre>	<pre> F77 PROGRAM MAIN WRITE(6,'(A)') ' A new line is implicit in Fortran' END </pre>
---	---

Floating Point

<pre> C main() { printf("%f",3.1459); printf("\n"); printf("%f\n",3.1459); printf("%10.10f\n",3.1459); printf("%.10f\n",3.1459); printf("%10f\n",3.1459); printf("%10f\n",3.1459e12); printf("\n"); printf("%e\n",3.1459); printf("%10.3e\n",3.1459); printf("%-10.3e\n",3.1459); printf("%+12.3e\n",3.1459); printf("%+-12.3e\n",3.1459); printf("%.10e\n",3.1459); printf("%10e\n",3.1459); printf("\n"); } </pre>	<pre> F77 PROGRAM MAIN WRITE(6,'(F10.3)') 3.1459 WRITE(6,'(E10.3)') 3.1459 WRITE(6,'(E20.12)') 3.1459 WRITE(6,'(1PE20.12)') 3.1459 END </pre>
---	---

```

printf("%E\n",3.1459);
printf("%10.3E\n",3.1459);
printf("%.10E\n",3.1459);
printf("%10E\n",3.1459);
printf("\n");

printf("%g\n",3.1459);
printf("%10.3g\n",3.1459);
printf("%.10g\n",3.1459);
printf("%10g\n",3.1459);
printf("\n");

}

```

C Conditional

```

#include <stdio.h>
#include <errno.h>

main(){

    int i = 0;

    /* == returns logical result, proper conditional */
    if ( i == 0 ) {
        printf("test 1: i is set to %d\n",i);
    }

    /* Using = not == for equality tests is a common typing mistake in C
    */
    /* Using = is valid syntax but it doesn't mean what you think it means!
    */
    /* i = 0 is false ( it returns 0 ) by definition, irrespective of the
    */
    /* value of i.
    */
    if ( i = 0 ) {
        printf("test 2: i is set to %d\n",i);
    }

    /* i = 1 is true ( it returns non-zero ) by definition, irrespective of
    */
    /* the value of i.
    */
    if ( i = 1 ) {
        printf("test 3: i is set to %d\n",i);
    }

}

```

F77 Conditional

```
PROGRAM MAIN

INTEGER I

I = 0

IF ( I .EQ. 0 ) THEN
  WRITE(6,'(A,I4)') 'I is set to', I
ENDIF

END
```