Rational and Logical Operators

== Equal to

!= Not equal to

> Greater than

>= Greater than or equal to

< Less than

<= Less than or equal to

&& And

|| Or

! Not

Bitwise Operators

& And

| Or

^ Xor

~ Not

>> Right shift

<< Left shift

ISR

```
#proagma code high_vector = 0x08
void high_interrupt (void){
        _asm goto high_ISR _endasm
}
#pragma code
#pragma interrupt high_ISR
void high_ISR (void) {
// ISR here
}
```

PIC Libraries

<delays.h>
Delay1TCY(n);
Delay10TCYx(n);
Delay100TCYx(n);
Delay1KTCYx(n);
Delay10KTCYx(n);

<adc.h>
<capture.h>
<ic2.h>
<portb.h>
<mwire.h>
<pwm.h>
<spi.h>
<timers.h>
<usart.h>

Other

```
#include <p18cxxx.h>
#define  label   constant
#pragma code [overlay] [section-name] [location]
_asm … _endasm
```

Control Flow

If

if(*expression*)
    statement;

If-else

if(*expression*)
    statement$_1$;
else
    satement$_2$;

Else-if

if(*expression*$_1$)
    statement$_1$;
else if (*expression*$_2$)
    satement$_2$;
else
    statement$_3$;

For

for (*expr*$_1$;*expr*$_2$;*expr*$_3$)
    statement;

While

while (*expression*)
    statement;

Switch

switch (*expression*){
    case *const_expr*$_1$:
        statement$_1$;
        break;
    case *const_expr*$_2$:
        statement$_2$;
        break;
    …
    default:
        statement$_n$;
}

Do-While

do
    statement;
while (*expression*);

examples

```
if (a > b)
        sum+=2;
```

```
if (a != 0)
        r = b;
else
        r = c;
```

```
if (abc > 0) return 5;
else if (abc == 0) return 0;
else return –5;
```

```
sum =0;
for (i=1; i<20; i++)
        sum = sum+1 * i;
```

```
int_cnt = 5;
while (int_cnt);
```

```
switch (i) {
        case 1: printf("*");
                break;
        case 2: printf("**");
                break;
        case 3: printf("***");
                break;
        case 4: printf("****");
                break;
        case 5: printf("*****");
        default:
                printf("\n");
```

```
int digit = 9;
do
        printf(%d ",digit—);
while (digit >=1);
```