



福州大学至诚学院
FUZHOU UNIVERSITY ZHICHENG COLLEGE

高级语言程序设计 (C语言与数据结构)

杨雄

83789047@qq.com

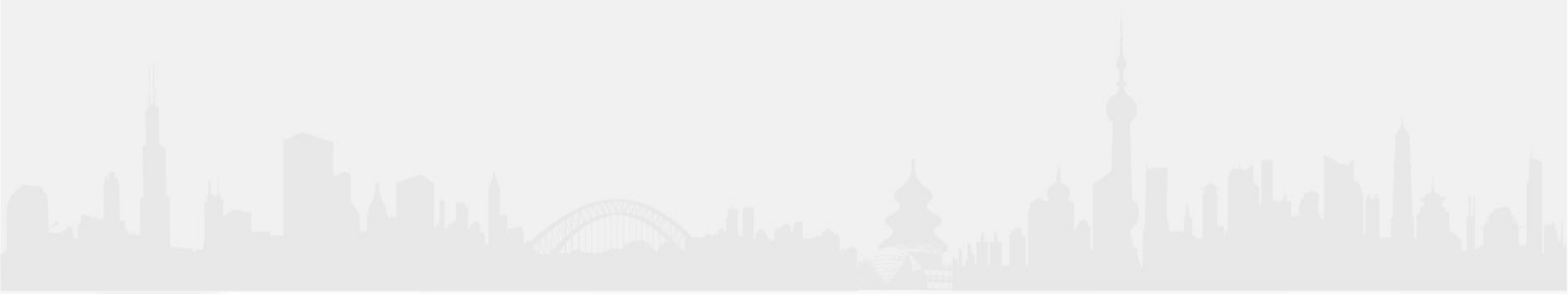




第四章 选择结构

4.1 用if语句实现选择结构

4.2 用switch语句实现多分支选择结构

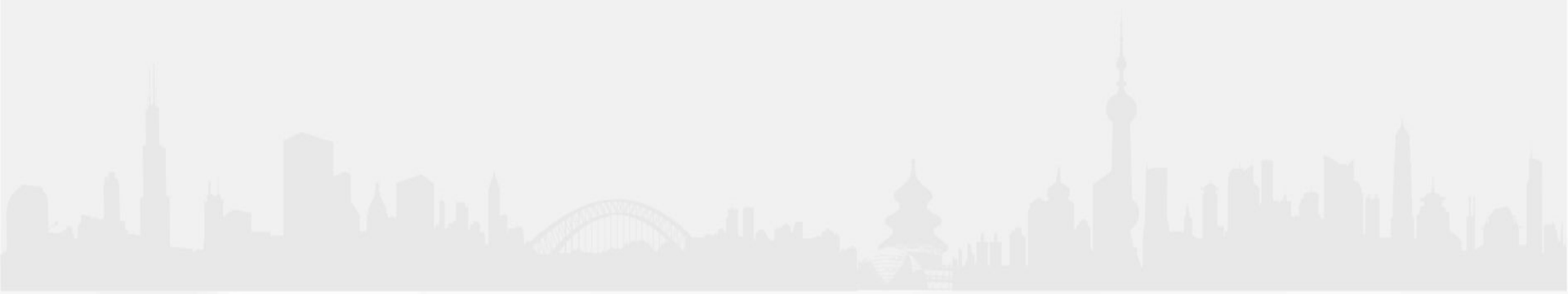


学习目的



Part.1

4.1 if语句



4.1.1 if语句的三种形式

1. 单分支结构(if 语句)

- 格式:

- **if (<表达式>)** 语句1 //语句1, **只一条语句**。

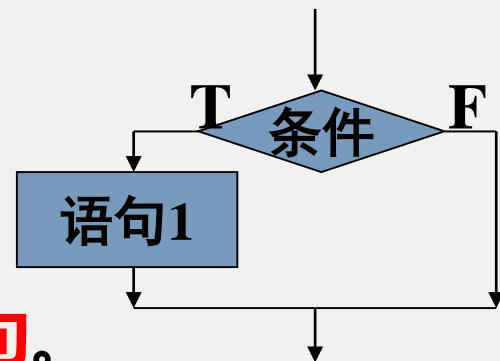
如果表达式为真, 执行语句1; 否则什么都不做;

- **if (<表达式>)**

- {

- }

- 例: `if (x > 100) printf("%d\n" , x);`
`y = y + 5 ;`



4.1.1 if语句的三种形式

例1： 输入三个数a, b, c, 选出最大数输出。

```
#include<stdio.h>
void main( )
{ float  a, b, c, big ;
  printf( “按格式%%f%%f%%f送数:\n” );
  scanf( “%f%f%f ” , &a, &b,&c);
  big=a ;
  if (b>big) big=b;
  if (c>big) big=c;
  printf( “最大数是: %.2f\n” , big);
}
```

4.1.1 if语句的三种形式

例2：执行下列程序后，m的值是多少？

```
#include<stdio.h>
void main()
{ float x, m ;
  scanf( "%f" , &x);
  if(x>=10.) m=2*x+1.0;
  if(x>=5.0) m=x+6.0;
  if(x>=0.0) m=10.4 ;
  printf( "%.0f\n" , m );
}
```

输入：12.6 ✓

输出m的值：

10

4.1.1 if语句的三种形式

2. 双分支结构 if... else语句

- 格式:

```
if (<表达式>
    语句1;
else
    语句2;
```

- 其中: 语句1称为if子句, 语句2称为else子句。
- 如果表达式为真, 执行语句1; 否则执行语句2
- else子句是if语句的一部分, 它不能作为语句单独使用, 必须与if配对使用。

4.1.1 if语句的三种形式

例3 输入一个整数,判断它是奇数还是偶数并打印出来。

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

```
void main()
```

```
{ int n;
```

```
scanf("%d", &n);
```

```
if (n%2==0)
```

```
printf("%d 是偶数\n", n);
```

```
else
```

```
printf("%d 是奇数\n", n);
```

```
}
```

1. 定义变量n

2. 给变量n赋初值

3. 判断n是否为偶数, 成立
输出n是偶数, 否则奇数。

4.1.1 if语句的三种形式

3. 多分支结构 if ... else if 语句

- 格式:

```
if (表达式1) 语句1;  
else if ( 表达式2 ) 语句2;  
else if ( 表达式3 ) 语句3;  
    ⋮  
else if ( 表达式n-1 )语句n-1;  
else      语句n;
```

4.1.1 if语句的三种形式

例4 输入某同学的成绩, 若 ≥ 90 分, 输出“A”, 若80–89为“B”, 若70–79为“C”, 若60–69为“D”, 若 < 60 分输出“不及格”。

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

```
void main ( )
```

```
{ float x ; printf ( “请按%f输入成绩:\n” );
```

```
scanf ( “%f” , &x );
```

```
if ( x >= 90 ) printf ( “A\n” );
```

```
else if ( x >= 80 ) printf ( “B\n” );
```

```
else if ( x >= 70 ) printf ( “C\n” );
```

```
else if ( x >= 60 ) printf ( “D\n” );
```

```
else printf ( “不及格\n” );
```

```
}
```

请按%f输入成绩:

55 ↙

不及格

4.1.1 if语句的三种形式

例4 实现此函数

$$y = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1 & x < 0 \end{cases}$$

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

```
void main( )
```

```
{ int x,y;           //变量x,y为整数类型
```

```
    printf( "x=" ); //提示信息
```

```
    scanf( "%d" ,&x); //键盘输入一个整数放入x中
```

```
    if (x>0)y=1;      //如果x>0, 则将1赋给y
```

```
    else if (x==0) y=0; //否则, 如果x==0, 则将0赋给y
```

```
    else y=-1;        //否则, 将-1赋给y
```

```
    printf( "y=%d\n" , y); //输出y的值
```

```
}
```

4.1.1 if语句的三种形式

三种格式的说明：

- 表达式一般为逻辑表达式或关系表达式, 也可以是任意的数值类型。

如: ① `if (3) printf ("oK");`

② `if ('a') printf ("%d" , 'a');`

- else子句不能脱离if子句而存在, 下面写法是错误的。

```
if (x >= 0) ;
```

```
else
```

```
    x = -x ;
```

4.1.2 if语句的嵌套

- 在if 语句中又包含有一个或多个if语句称为 if 语句的嵌套。

格式一 **if (表达式1)**
 if (表达式2) 语句1

例: **if (a > 0)**
 if (b > 0)
 printf("%d,%d\n" , a, b);

4.1.2 if语句的嵌套

格式二:

```
if ( )  
    if ( ) 语句1  
    else 语句2  
else  
    if ( ) 语句3  
    else 语句4
```


4.1.2 if语句的嵌套

说明:

- else与if必须成对出现, 且else总是与最近的一个未配对的if配对。如:

```
if ( )
```

```
    if ( ) 语句1
```

```
else
```

```
    if ( ) 语句2
```

```
    else 语句3
```



```
if ( )
```

```
    if ( ) 语句1
```

```
    else
```

```
        if ( ) 语句2
```

```
        else 语句3
```

4.1.2 if语句的嵌套

说明:

- 如果 if与else的数目不同, 为实现指定的操作, 可以加花括号来确定配对关系。

如:

```
if ( )  
{  
    { if ( ) 语句1}  
    else  
        语句2  
}
```

4.1.2 if语句的嵌套

说明:

- 内层的选择结构必须完整地嵌套在外层的选择结构内, 两者不允许交叉。



- 为了表示嵌套关系, 程序书写时应采用缩进格式, 使程序层次分明, 易于理解。

4.1.2 if语句的嵌套

例6

```
void main( )  
{ int a=2, b=1, c=2;  
  if (a)  
    if ( b<0 ) c=0;  
    else c++ ;  
  printf( "%d\n" ,c );  
}
```

输出结果:

3

```
void main( )  
{ int a=2, b=1, c=2;  
  if (a)  
    { if (b<0) c=0; }  
    else c++ ;  
  printf( "%d\n" ,c );  
}
```

输出结果:

2

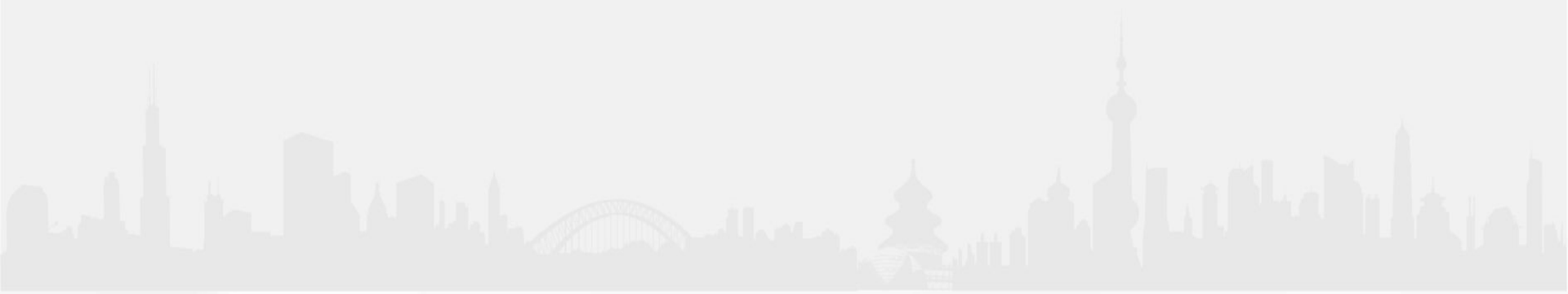
4.1.2 if语句的嵌套

例7 任意输入一个字符, 判断它是字母还是数字。

```
#include <stdio.h>
void main()
{ char c=getchar();
  if ((c>= 'A' && c<= 'Z' )||(c>= 'a' && c<= 'z' ))
    printf( "%c 是一个字母。" , c );
  else if (c>= '0' && c<= '9' )
    printf( "%c 是一个数字。" , c );
  else
    printf( "%c 是其他字符。" , c );
}
```

Part.2

4.2 switch语句



4.2 switch语句

- switch语句是多分支选择语句，也称开关语句。

格式：

switch(表达式)

{ **case** 常量表达式1: 语句组1; [**break**];

case 常量表达式2: 语句组2; [**break**];

.....

case 常量表达式n: 语句组n; [**break**];

default : 语句n+1;

}

- 若表达式 == 常量表达式，则执行语句;
- 若表达式 != 常量表达式，则执行default后面的语句;

4.2 switch语句

例: 根据考试成绩输出百分制分数段

```
char grade=getchar();  
switch(grade)  
{ case 'A' : printf( "85~100\n" );  
  case 'B' : printf( "70~84\n" );  
  case 'C' : printf( "60~69\n" );  
  case 'D' : printf( "<60\n" );  
  default : printf( "error\n" );  
}
```

若 grade=' B'
输出: 70~84
60~69
<60
error

4.2 switch语句

改进后的代码:

```
char grade= 'B' ;  
switch(grade)  
{ case 'A' : printf( "85~100\n" ); break;  
  case 'B' : printf( "70~84\n" ); break;  
  case 'C' : printf( "60~69\n" ); break;  
  case 'D' : printf( "<60\n" ); break;  
    : printf( "error\n" );  
}
```

break语句终止
switch语句的执行

70~84

4.2 switch语句

说明：

- switch后表达式的类型与case后面常量值的类型必须匹配。
且只能是**整型、字符型或枚举型表达式**。

例:

```
float f=4.0 ;  
swtch( f )    /*error */  
{  
    ...  
}
```

- 每个case后的常数表达式的值**不允许相同**, 如:

```
case 'A' : printf( "this is A\n" );  
case 65  : printf( "this is 65\n" );  
//error, 'A' 等值于65
```

4.2 switch语句

说明：

- 各个case包括(default)的出现次序可以任意。在每个case分支都带有break的情况下，case次序不影响执行结果。

```
char grade= 'E' ;  
switch(grade)  
{ case 'C' : printf( "60~69\n" ); break;  
  default : printf( "error\n" ); break;  
  case 'D' : printf( "<60\n" ); break;  
  case 'A' : printf( "85~100\n" ); break;  
  case 'B' : printf( "70~84\n" );  
} 输出结果: error
```

4.2 switch语句

- 多个case语句可以共用一组执行语句。如：

```
switch(n)
{ case 1:   case 3:
  case 5:   case 7:
  case 9:   printf( "奇数 \n" ); break ;
  case 0:   case 2:
  case 4:   case 6:
  case 8:   printf( "偶数 \n" ); break ;
}
```

该switch语句中，当n=1,3,5,7,9时，输出“奇数”；
当n=0,2,4,6,8时，输出“偶数”。

习 题

1. C语言中，逻辑“真”等价于 C。

- a) 大于0的数
- b) 大于0的整数
- c) 非0的数
- d) 非0的整数

2. C语言中规定,else总是与D的 if 组成配对关系。

- a) 缩进距离相等
- b) 前面一行上
- c) 同一行上
- d) 在它前面距离最近且未与其它else配对

习 题

3. 以下程序输出结果是:

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

```
void main( )
```

```
{ int  a=-1, b=4, k ;
```

```
    k=(++a<0) && !(b--<=0);
```

```
    printf( "%d %d %d\n" , k, a, b);
```

```
}
```

a) 1 0 4

b) 1 0 3

c) 0 0 3

d) 0 0 4



习 题

4. 两次运行下面程序, 如果从键盘上分别输入6和5, 则输出结果是:

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

```
void main( )
```

```
{ int m ;
```

```
    scanf(" %d" , &m) ;
```

```
    if (m++ > 5) printf( "%d\n" , m);
```

```
    else printf( " %d\n" , m--);
```

```
}
```

✓ A) 7和6 B) 6和4 C) 7和5 D) 6和5

习 题

5. 阅读程序，写出下列程序执行后的结果。

```
void main( )  
{ int  a=-1, b=3, c=3, s, w, t ;  s=w=t=0 ;  
  if (c>0)  
    s=a+b;  
  if ( a<=0 )  
    { if (b>0)  
      if (c<=0) w=a-b; }  
  else  
    if(c>0) w=a-b;  
    else    t=c;  
  printf( "%d, %d, %d\n" , s, w, t );  
}
```

输出结果:

2,0,0

习 题

6. 写出下列程序执行后的结果

```
#include <stdio.h>
void main( )
{ int  a, b, c, d, x;
  a=c=x=0;  b=1;  d=20;
  if (a)
    d=d-10;
  else
    if (!b)
      if(!c)
        x=15;
      else
        x=25;
  printf( "%d,%d\n" , d, x);
}
```

输出结果：
20,0

习 题

7. 有如下程序段，则输出结果是：

```
int k=5, m=1;  
switch(k)  
{ case 1: m++;  
  case 5: m*=3;  
  case 9: m+=4; break;  
  case 11: m+=1;  
  default : m+=3; }  
printf( "%d\n" ,m);
```

- A) 10 B) 7 C) 3 D) 11
- ✓

习 题

8. 以下程序段的运行结果是 **【*1**3*】**

```
int x=1, y=0;
```

```
switch(x)
```

```
{case 1:
```

```
    switch (y)
```

```
    { case 0: printf( “*1*” ); break;
```

```
      case 1: printf( “*2*” ); break;
```

```
    }
```

```
    case 2: printf( “*3*” ); break;
```

```
}
```

习 题

9. 写出下面程序输出结果 【 3 】 .

```
#include <stdio.h>
void main()
{ int a=13, b=21, m=0;
  switch( a%3 )
  { case 0: m++;break;
    case 1: m++;
      switch( b%2 )
      {default: m++; break;
        case 0: m++; break; }
    }
  printf( "%d\n" , m) ;
}
```

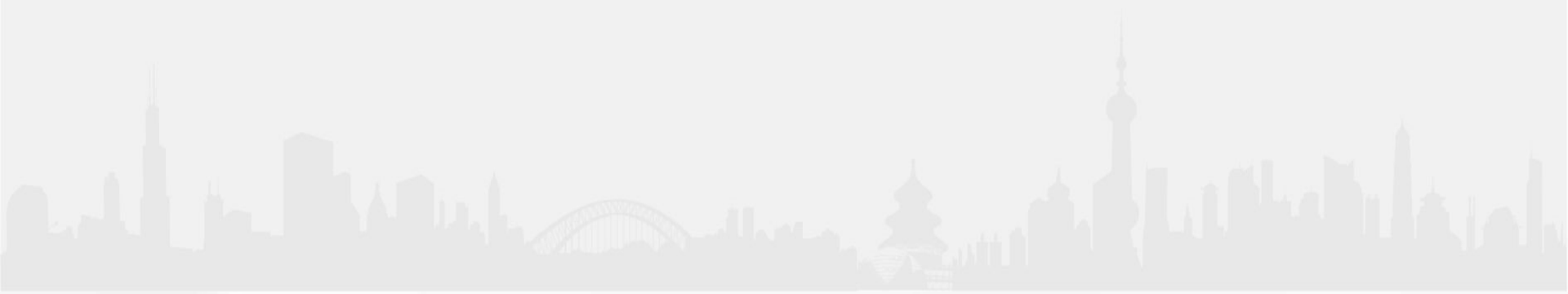
习题

10. 当 $a=1, b=3, c=5, d=4$ 时, 执行下面一段程序后,
x的值为: A) 1 **B) 2** C) 3 D) 6

```
if (a < b)
{
    if (c < d) x = 1;
    else
    {
        if (a < c)
        {
            if (b < d) x = 2;
            else x = 3;
        }
        else x = 6;
    }
    else x = 7;
```


Part.3

总结



总 结

```
if (表达式1) 语句1;  
else if ( 表达式2 ) 语句2;  
else if ( 表达式3 ) 语句3;  
    ⋮  
else if ( 表达式n-1 )语句n-1;  
else      语句n;
```

总 结

switch(表达式)

```
{ case 常量表达式1: 语句组1; [break;]
  case 常量表达式2: 语句组2; [break;]
    .....
  case 常量表达式n: 语句组n; [break;]
  default : 语句n+1;
}
```

- 若表达式 == 常量表达式, 则执行语句;
- 若表达式 != 常量表达式, 则执行default后面的语句;