强制类型转换_百度百科

强制类型转换编辑

当操作数的类型不同，而且不属于基本数据类型时，经常需要将操作数转化为所需要的类型，这个过程即为强制类型转换。强制类型转换具有两种形式：显式强制转换和隐式强制类型转换。

中文名
强制类型转换
形　式
显式强制转换和隐式强制类型转换
类　型
数据类型
运用语言
C语言

目录

强制类型转换

编辑

1、显式强制类型转换

C中显式强制类型转换很简单，格式如下：
TYPE b = (TYPE) a;
其中，TYPE为类型描述符，如int，float等。经强制类型转换运算符运算后，返回一个具有TYPE类型的数值，这种强制类型转换操作并不改变操作数本身，运算后操作数本身未改变，例如：
int n=0xab65;
char a=（char）n;
上述强制类型转换的结果是将整型值0xab65的高端一个字节删掉，将低端一个字节的内容作为char型数值赋值给变量a，而经过类型转换后n的值并未改变。
C++中强制类型转换函数有4个：const_cast(用于去除const属性），static_cast(用于基本类型的强制转换），dynamic_cast(用于多态类型之间的类型转换），reinterpreter_cast(用于不同类型之间的指针之间的转换，最常用的就是不同类型之间函数指针的转换）。

2、隐式强制类型转换

隐式类型转换发生在赋值表达式和有返回值的函数调用表达式中。在赋值表达式中，如果赋值符左右两侧的操作数类型不同，则将赋值符右边操作数强制转换为赋值符左侧的类型数值后，赋值给赋值符左侧的变量。在函数调用时，如果return后面表达式的类型与函数返回值类型不同，则在返回值时将return后面表达式的数值强制转换为函数返回值类型后，再将值返回，如：
int n;
double d=3.88;
n=d；//执行本句后，n的值为3，而d的值仍是3.88。

自动类型转换

编辑

在C语言中，自动类型转换遵循以下规则：
1、若参与运算量的类型不同，则先转换成同一类型，然后进行运算。
2、转换按数据长度增加的方向进行，以保证精度不降低。如int型和long型运算时，先把int量转成long型后再进行运算。
a、若两种类型的字节数不同，转换成字节数高的类型
b、若两种类型的字节数相同，且一种有符号，一种无符号，则转换成无符号类型

3、所有的浮点运算都是以双精度进行的，即使仅含float单精度量运算的表达式，也要先转换成double型，再作运算。
4、char型和short型（在visual c++等环境下）参与运算时，必须先转换成int型。
5、在赋值运算中，赋值号两边量的数据类型不同时，赋值号右边量的类型将转换为左边量的类型。如果右边量的数据类型长度比左边长时，将丢失一部分数据，这样会降低精度，丢失的部分直接舍去。

隐式自动类型转换

编辑

C++语言编译系统提供的内部数据类型的隐式自动转换规则如下：
1、执行算术运算时，低类型(短字节)可以转换为高类型(长字节)；例如： int型转换成double型，char型转换成int型等等；
2、赋值表达式中，等号右边表达式的值的类型自动隐式地转换为左边变量的类型，并赋值给它；
3、函数调用时，将实参的值传递给形参，系统首先会自动隐式地把实参的值的类型转换为形参的类型，然后再赋值给形参；
4、函数有返回值时，系统首先会自动隐式地将返回表达式的值的类型转换为函数的返回类型，然后再赋值给调用函数返回；

单参构造函数的类型转换功能

编辑

实际应用中，当一个类定义中提供了单个参数的构造函数时，该类便提供了一种将其它数据类型的数值或变量转换为用户所定义数据类型的方法；因此，可以说，单个参数的构造函数提供了数据类型转换的功能。