C++中，如果两个类型可以相互转换，则称这两个类型相关。

由编译器自动执行的转换称为隐式类型转换。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 文件名 ：test.cpp

/\* 作者 ：刘珅珅

/\* 日期 ：2011/11/2

/\* 功能描述 ：隐式类型的转换测试 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

using namespace std;

int main(int argc, char \*argv[])

{

bool flag;

char cval;

short sval;

unsigned short usval;

int ival;

unsigned int uival;

long lval;

unsigned long ulval;

float fval;

double dval;

3.14159L + 'a'; // 将字符a提升为int，然后转换为long double

dval + ival; // ival转换为double

dval + fval; // fval转换为double

ival = dval; // dval被截断转换为int

flag = dval; // 如果dval是0，flag是false，否则是true

cval + fval; // cval提升为int，然后从int转换为float

sval + cval; // sval和cval都提升为int

cval + lval; // cval转换为long

ival + ulval; // ival转换为unsigned long

// 提升依赖于unsigned short和int的尺寸，如果int型足够表示所有的unsigned short型的值

// 则将unsigned short提升为int，否则两个都提升为unsigned int

usval + ival;

// 提升依赖于unsigned int和long的尺寸，如果long型足够表示所有的unsigned int型的值

//则将unsigned int提升为long，否则两个都提升为unsigned long

uival + lval;

return 0;

}

强制类型转换（cast）

强制类型转换在本质上是非常危险的，能不用最好不用。

强制类型转换的操作符有：static\_cast，dynamic\_cast，const\_cast，reinterpret\_cast。

命名的强制类型转换的操作：

cast-name<type>(expression);

其中cast-name为操作符的一种，type为转换的目标类型，expression为被强制转换的值。例：static\_cast<int>(dval);

旧式强制类型转换：

char \*pc = (char\*) ip;

旧式强制类型转换的两种形式：

type(expr);

(type)expr;

如果使用强制类型转换最好不要使用旧式强制类型转换，而应该使用强制类型转换操作符。