1. c++中有哪4个与类型转换相关的关键字？这些关键字各有什么特点，应该在什么场合使用？

答：static\_cast:可以实现C++中内置基本数据类型之间的相互转换，enum、struct、 int、char、float等。它不能进行无关类型(如非基类和子类)指针之间的转换。

int c=static\_cast<int>(7.987);

如果涉及到类的话，static\_cast只能在有**相互联系的类型**中进行相互转换,不一定包含虚函数。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | class A  {};  class B:public A  {};  class C  {};    int main()  {      A\* a=new A;      B\* b;      C\* c;      b=static\_cast<B>(a);  // 编译不会报错, B类继承A类      c=static\_cast<B>(a);  // 编译报错, C类与A类没有任何关系      return 1; |

const\_cast: const\_cast操作不能在不同的种类间转换。相反，它仅仅把一个它作用的表达式转换成常量。它可以使一个本来不是const类型的数据转换成const类型的，或者把const属性去掉。

reinterpret\_cast: （interpret是解释的意思，reinterpret即为重新解释，此标识符的意思即为数据的二进制形式重新解释，但是不改变其值。）有着和C风格的强制转换同样的能力。它可以转化任何内置的数据类型为其他任何的数据类型，也可以转化任何指针类型为其他的类型。它甚至可以转化内置的数据类型为指针，无须考虑类型安全或者常量的情形。不到万不得已绝对不用。

dynamic\_cast主要用于类层次间的上行转换和下行转换，还可以用于类之间的交叉转换。

在类层次间进行上行转换时，dynamic\_cast和static\_cast的效果是一样的；

在进行下行转换时，dynamic\_cast具有类型检查的功能，比static\_cast更安全。

https://www.cnblogs.com/rednodel/p/5800142.html

1. 虚函数？

答：虚函数的作用是允许在派生类中重新定义与基类同名的函数，并且可以通过基类指针或引用来访问基类和派生类中的同名函数。由虚函数实现的动态多态性就是：同一类族中不同类的对象，对同一函数调用作出不同的响应。<https://www.cnblogs.com/jianyungsun/p/6361670.html>

1. 定义一个空的类型，里面没有任何成员变量和成员函数。对该类型求sizeof,结果为？
   1. 答：结果为1，声明该类型的实例时，必须在内存中占有一定的空间，否则无法使用这些实例。至于占用多少内存，由编译器决定。在Visual Studio中，每个空类型的实例占用1字节的空间。
   2. 若添加一个构造函数和析构函数，再求sizeof呢？答：仍为1，调用构造和析构只需要知道函数地址即可，地址只与类型相关，与类型实例无关，不会在实例中添加额外的信息。
   3. 将析构函数标记为虚函数呢？答：C++的编译器一旦发现一个类型中有虚函数，就会为该类型生成虚函数表，并在该类型的每一个实例中添加一个指向虚函数表的指针。在32位机器上，一个指针占4字节空间，求sizeof得到4，如果是64位的机器，则占8字节，得8。
2. C++标准不允许拷贝构造函数传值参数，会无休止递归调用导致栈溢出。应把传值参数改成常量引用。
3. 三种访问权限：public:可以被任意实体访问，protected:只允许子类及本类的成员函数访问
4. private:只允许本类的成员函数访问