

C 类型转换运算符的使用方法 PDF转换可能丢失图片或格式
，建议阅读原文

https://www.100test.com/kao_ti2020/285/2021_2022_C___E7_B1_BB_E5_9E_8B_E8_c97_285148.htm

C 提供了四个新的类型转换运算符：`const_cast` `dynamic_cast` `reinterpret_cast` `static_cast` 使用方法：

`cast_operator (object)` 类型转换操作符 要转换的类型 要进行转换的对象

`dynamic_cast` 将一个基类引用或指针转换为一个派生类应用或指针，或者将一个派生类引用或指针转换为一个基类引用或指针。例：`class Shape { ... }. class Circle : public Shape { ... }. Shape *sp ; Circle *cp = dynamic_cast (sp).`

`class Control { ... }. class TextBox : public Control { ... }. Control ampctl = dynamic_cast (cr).` `static_cast` 不局限于同一多态类层次中的基类和派生类，可使用`static_cast`调用处于不同层次的类型之间的隐式转换。例：`class B { ... }. class D : public B { ... }. void f(B* pb, D* pd) { D* pd2 = static_cast(pb). // not safe, pb may // point to just B B* pb2 = static_cast(pd). // safe conversion ... }`

`reinterpret_cast` 将指针类型转换为其他指针类型，将数字转换为指针或将指针转换为数字。例：`void * getmen() { static char buf[100]. ... return buf. } ... char *cp = reinterpret_cast (getmen).`

`const_cast` 移去对象的`const`, `volatile`, 和 `__unaligned` 属性。例：`class CCTest { public: void setNumber(int). void printNumber() const. private: int number. }. void`

`CCTest::setNumber(int num) { number = num. } void CCTest::printNumber() const { cout const_cast(this)->number--.`

`cout }` 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com