例：程序C++11\_test13

class Base {

public:

Base(int number) : number\_(number)

{

*cout* << "Construct Base" << *endl*;

}

Base(const Base& org) : number\_(org.number\_)

{

*cout* << "Copy construct Base" << *endl*;

}

Base(Base&& org) : number\_(org.number\_)

{

*cout* << "Move construct Base" << *endl*;

}

virtual ~Base()

{

*cout* << "Destruct Base" << *endl*;

}

private:

int number\_;

};

class Derived : Base

{

public:

Derived(int no) : Base(no), no\_(no)

{

*cout* << "Construct Derived" << *endl*;

}

Derived(const Derived& org) : Base(org), no\_(org.no\_)

{

*cout* << "Copy construct Derived" << *endl*;

}

Derived(Derived&& org) : Base(std::*move*(org)), no\_(org.no\_)

{

*cout* << "Move construct Derived" << *endl*;

}

~Derived()

{

*cout* << "Destruct Derived" << *endl*;

}

private:

int no\_;

};

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

{

Derived d1(3);

*cout* << *endl*;

Derived d2(d1);

*cout* << *endl*;

Derived d3(*move*(d1));

*cout* << *endl*;

return 0;

}

输出为：

Construct Base

Construct Derived

Copy construct Base

Copy construct Derived

Move construct Base

Move construct Derived

Destruct Derived

Destruct Base

Destruct Derived

Destruct Base

Destruct Derived