友元与继承：

友元关系不能继承。基类的友元对派生类自有的成员没有特殊访问权限，但可以访问派生类继承自基类的成员。同样的，派生类的友元也无法访问基类的private成员和protected成员，也无法访问派生类继承自基类的private成员和protected成员，它只能访问派生类自有的成员。另外，如果类A是类B的友元，而类C是类A的继承，但类C无法访问类B的private成员。见下例：

例：程序class\_test18

class Base

{

friend class Frnd; // Frnd为Base的友元

private:

int number;

};

class Derive : public Base

{

friend class Frnd1; // Frnd1为Derive的友元

private:

int count;

};

class Frnd

{

public:

int mem(Base b)

{

return b.number; // ok

}

int mem(Derive d)

{

// return d.count; // error，基类的友元无法访问Derive类自有的private成员

return d.number; // ok，基类的友元可以访问Derive类继承自基类Base的private成员

}

};

class Derive1 : public Frnd

{

public:

int mem(Base b)

{

// 即使基类Frnd是Base类的友元，但其派生类Derive1也无法访问Base类的private成员

// return b.number; // error无法访问Base类的private成员

return 0;

}

class Frnd1

{

public:

int mem(Base b)

{

// return b.number; // error，派生类Derive的友元Frnd1无法访问基类Base的private成员

return 0;

}

int mem(Derive d)

{

return d.count; // ok，派生类Derive的友元Frnd1可以访问Derive类自有的private成员

// return d.number; // error，派生类Derive的友元Frnd1无法访问Derive类继承自基类Base的private成员

}

};

// 静态成员与继承

例：程序class\_test19

#include <iostream>

class Base

{

public:

static void func()

{

std::cout << "Base的静态函数" << std::endl;

}

protected:

int number;

static int num\_sta;

};

class Derive : public Base

{

public:

void f()

{

Base::func(); // 调用基类的静态成员，基类的静态成员函数可以和派生类的函数相同。

Base::num\_sta = 5;

num\_sta = 6;

func();

}

static void func()

{

std::cout << "Derive的静态函数" << std::endl;

}

private:

static int num\_sta;

};

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

{

Derive d;

d.f();

return 0;

}

结果：

Base的静态函数

Derive的静态函数