

Supplement IV.A: Multiple Inheritance

For Introduction to C++ Programming
By Y. Daniel Liang

C++ allows you to derive a class from multiple base classes. This is known as *multiple inheritance*. Listing 1 gives an example.

Listing 1 MultipleInheritanceDemo.cpp

```
#include <iostream>
using namespace std;

class B1
{
public:
    void p1()
    {
        cout << "from B1" << endl;
    }
};

class B2
{
public:
    void p2()
    {
        cout << "from B2" << endl;
    }
};

class A: public B1, public B2
{
public:
    void p3()
    {
        cout << "from A" << endl;
    }
};

int main()
{
    A a;
    a.p1();
    a.p2();
    a.p3();
    return 0;
}
```

Class **A** derives from classes **B1** and **B2** (line 22). An object **a** of class **A** is created in line 24. Function **p1** is inherited to **A** from **B1** and function **p2** is inherited to **A** from **B2**.

Multiple inheritance is a powerful capability to develop reusable software. But it causes ambiguity in some cases. For example, if you change the function name **p2** in **B2** to **p1**, the compiler will report an ambiguity error. To resolve this ambiguity, redefine function **p1** in class **A**.