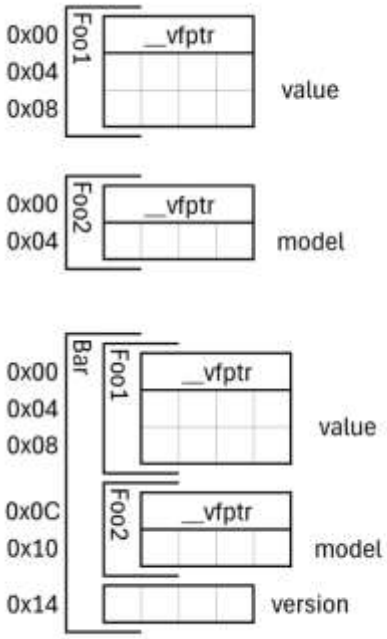
```cpp
#include <iostream>
using std::cout;

struct Foo1
{
   virtual ~Foo1() {}
   double value{};

   void Func1() {
      cout << "Foo1::Func1() this = " << this << '\n';
      FuncB();
   }
   virtual void FuncB() {
      cout << "Foo1::FuncB() this = " << this << '\n';
   }
};
struct Foo2
{
   virtual ~Foo2() {}
   int model{};

   void Func2() {
      cout << "Foo2::Func2() this = " << this << '\n';
      FuncB();
   }
   virtual void FuncB() {
      cout << "Foo2::FuncB() this = " << this << '\n';
   }
};
struct Bar : public Foo1, Foo2
{
   int version{};
   void FuncA() {
      cout << "Bar ::FuncA() this = " << this << '\n';
      Func1();
      Func2();
   }
   virtual void FuncB() {
      cout << "Bar ::FuncB() this = " << this << '\n';
      Foo1::FuncB();
      Foo2::FuncB();
   }
};

int main()
{
   Bar bar;
   cout << "&bar = " <<  &bar << '\n';
   cout << "&bar::Foo1 = " <<  dynamic_cast<Foo1*>(&bar) << '\n';
   cout << "&bar::Foo2 = " <<  dynamic_cast<Foo2*>(&bar) << '\n';
   bar.FuncA();
};
```




```
Foo1::`vftable'= {
    Foo1::'vector deleing destructor'(uint),
    Foo1::FuncB(void),
};
Foo2::`vftable'= {
    Foo2::`vector deleing destructor'(uint),
    Foo2::FuncB(void),
};

Bar::Foo1::`vftable'= {
    Bar::'vector deleing destructor'(uint),
    Bar::FuncB(void),
};
Bar::Foo2::`vftable'= {
    Bar::`vector deleing destructor'`adjustor{16}'(uint),
    Bar::FuncB'adjustor{16}'(void),
};
```