```cpp
//
//  main.cpp
//  AbsoluteCpp_ch15_7
//


//Program to illustrate use of a virtual function to defeat the slicing
 problem.
#include <string>
#include <iostream>
using std::string;
using std::cout;
using std::endl;

class Pet
{
public:
    string name;
    virtual void print( ) const;
};

class Dog : public Pet
{
public:
    string breed;
    virtual void print( ) const;
};

int main( )
{
    Dog vdog;
    Pet vpet;

    vdog.name = "Tiny";
    vdog.breed = "Great Dane";
    vpet = vdog;
    cout << "The slicing problem:\n";
    // vpet.breed; //is illegal since class Pet has no member named breed.
    vpet.print( );
    // Dog vdog1 = (Dog)vpet;
    // vdog1.print();
    cout << "Note that it was print from Pet that was invoked.\n";

    cout << "The slicing problem defeated:\n";
    Pet *ppet;
    Dog *pdog;
    pdog = new Dog;

    pdog->name = "Tiny";
    pdog->breed = "Great Dane";
    ppet = pdog;
    ppet->print( );
    pdog->print( );

    //The following, which accesses member variables directly
```

```
    //rather than via virtual functions would produce an error:
    //cout << "name: " << ppet->name << "  breed: "
    //     << ppet->breed << endl;
    //It generates an error message saying
    //class Pet has no member named breed.

    return 0;
}

void Dog::print( ) const
{
    cout << "name: " << name << endl;
    cout << "breed: " << breed << endl;
}

void Pet::print( ) const
{
    cout << "name: " << name << endl;
}
```