

```

//
//  main.cpp
//  AbsoluteCpp_ch15_1
//
//

//Demonstrates the performance of the virtual function bill.
#include <iostream>
#include "sale.h" //Not really needed, but safe due to ifndef.
#include "discountsale.h"
#include <vector>
using std::cout;
using std::endl;
using std::ios;
using std::vector;
using namespace SavitchSale;

int main( )
{

    Sale simple(10.00); //One item at $10.00.
    DiscountSale discount(11.00, 10); //One item at $11.00 with a 10%
        discount.

    cout << "---- test 1 ----" << endl;
    vector<Sale> sales;
    sales.push_back(simple);
    sales.push_back(discount);
    cout << sales[0].bill() << endl;    // 10
    cout << sales[1].bill() << endl;    // 11
    cout << discount.bill() << endl;    // 9.9

    cout << "---- test 2 ----" << endl;
    Sale * slaes2 = new Sale[2];
    slaes2[0] = simple;
    slaes2[1] = discount;
    cout << slaes2[0].bill() << endl;    // 10
    cout << slaes2[1].bill() << endl;    // 11
    cout << discount.bill() << endl;    // 9.9

    cout << "---- test 3 ----" << endl;
    Sale & saleRef = discount;
    cout << saleRef.bill() << endl;    // 10

    cout << "---- test 4 ----" << endl;
    Sale * salePtr;
    salePtr = &discount;
    cout << salePtr->bill() << endl;    // 9.9

    cout << "---- test 5 ----" << endl;
    Sale * salePtr2 = new Sale;
    salePtr2 = &discount;

```

```
cout << salePtr2->bill() << endl;    // 9.9

cout.setf(ios::fixed);
cout.setf(ios::showpoint);
cout.precision(2);

if (discount < simple)
{
    cout << "Discounted item is cheaper.\n";
    cout << "Savings is $" << simple.savings(discount) << endl;
}
else
    cout << "Discounted item is not cheaper.\n";

return 0;
}
```

```

//
//  main.cpp
//  AbsoluteCpp_ch15_1
//
//

//Demonstrates the performance of the virtual function bill.
#include <iostream>
#include "sale.h" //Not really needed, but safe due to ifndef.
#include "discountsale.h"
using std::cout;
using std::endl;
using std::ios;
using namespace SavitchSale;

int main( )
{
    Sale simple(10.00); //One item at $10.00.
    DiscountSale discount(11.00, 10); //One item at $11.00 with a 10%
        discount.

    cout.setf(ios::fixed);
    cout.setf(ios::showpoint);
    cout.precision(2);

    if (discount < simple)
    {
        cout << "Discounted item is cheaper.\n";
        cout << "Savings is $" << simple.savings(discount) << endl;
    }
    else
        cout << "Discounted item is not cheaper.\n";

    return 0;
}

```

```
//This is the header file sale.h.
//This is the interface for the class Sale.
//Sale is a class for simple sales.

#ifndef SALE_H
#define SALE_H

namespace SavitchSale
{

    class Sale
    {
    public:
        Sale( );
        Sale(double thePrice);
        double getPrice( ) const;
        void setPrice(double newPrice);
        virtual double bill( ) const;
        double savings(const Sale& other) const;
        //Returns the savings if you buy other instead of the calling
        //object.
    private:
        double price;
    };

    bool operator < (const Sale& first, const Sale& second);
    //Compares two sales to see which is larger.

} //SavitchSale

#endif // SALE_H
```

```
//This is the file discountsale.h.
//This is the interface for the class DiscountSale.

#ifndef DISCOUNTSALE_H
#define DISCOUNTSALE_H
#include "sale.h"

namespace SavitchSale
{

    class DiscountSale : public Sale
    {

    public:
        DiscountSale( );
        DiscountSale(double thePrice, double theDiscount);
        //Discount is expressed as a percent of the price.
        //A negative discount is a price increase.
        double getDiscount( ) const;
        void setDiscount(double newDiscount);
        double bill( ) const;
    private:
        double discount;

    };

} //SavitchSale

#endif //DISCOUNTSALE_H
```

```
//This is the file sale.cpp.  
//This is the implementation for the class Sale.  
//The interface for the class Sale is in the file sale.h.
```

```
#include <iostream>  
#include "sale.h"  
#include <cstdlib>  
using std::cout;  
  
namespace SavitchSale  
{  
  
    Sale::Sale( ) : price(0)  
    {  
        //Intentionally empty  
    }  
  
    Sale::Sale(double thePrice)  
    {  
        if (thePrice >= 0)  
            price = thePrice;  
        else  
        {  
            cout << "Error: Cannot have a negative price!\n";  
            exit(1);  
        }  
    }  
  
    double Sale::bill( ) const  
    {  
        return price;  
    }  
  
    double Sale::getPrice( ) const  
    {  
        return price;  
    }  
  
    void Sale::setPrice(double newPrice)  
    {  
        if (newPrice >= 0)  
            price = newPrice;  
        else  
        {  
            cout << "Error: Cannot have a negative price!\n";  
            exit(1);  
        }  
    }  
  
    double Sale::savings(const Sale& other) const  
    {  
        return (bill( ) - other.bill( ));  
    }  
}
```

```
bool operator < (const Sale& first, const Sale& second)
{
    return (first.bill( ) < second.bill( ));
}

} // SavitchSale
```

```

//This is the implementation for the class DiscountSale.
//This is the file discountsale.cpp.
//The interface for the class DiscountSale is in the header file
discountsale.h.
#include "discountsale.h"

namespace SavitchSale
{

    DiscountSale::DiscountSale( ) : Sale( ), discount(0)
    {
        //Intentionally empty
    }

    DiscountSale::DiscountSale(double thePrice, double theDiscount)
        : Sale(thePrice), discount(theDiscount)
    {
        //Intentionally empty
    }

    double DiscountSale::getDiscount( ) const
    {
        return discount;
    }

    void DiscountSale::setDiscount(double newDiscount)
    {
        discount = newDiscount;
    }

    double DiscountSale::bill( ) const
    {
        double fraction = discount/100;
        return (1 - fraction)*getPrice( );
    }

} //SavitchSale

```