```
// Thomas Gibbons
//October 10, 2016

#include <iostream>
#include <cstring>

class Stock{
    private:
        char company[30];
        int shares;
        double share_val;
        double total_val;
        void set_tot(){total_val = shares * share_val; }
        Stock acquire(const char* co);
        Stock acquire(const char* co, int n, double pr);
    public:
        // Public methods
        Stock buy(int num, double price);
        Stock sell(int num, double price);
        Stock update(double price);
        void display();
        // Constructors: same name as the class
        Stock();    // default, no arguments.
        Stock(const char*);
        Stock(const char*, int, double);
};

Stock::Stock(){
    //initialize empty stock
    company[0] = '\0';
    shares = 0;
    share_val = 0;
    set_tot();
}

Stock::Stock(const char* co){
    //initialize just the name of a stock
    acquire(co);
}

Stock::Stock(const char* co, int n, double pr){
    //initialize name, shares and price
    acquire(co, n, pr);
}

Stock Stock::buy(int num, double price){
    shares = shares + num;
    update(price);
    std::cout   << std::endl
                << num << " shares of " << company << " bought at $" << price << std::endl;
    return *this;
}
```

```
Stock Stock::sell(int num, double price){
    if( shares < num){
        std::cerr << "Not enough shares. Selling all shares";
        num=shares;
        shares = 0;
    }
    else{
        shares = shares - num;
    }
    update(price);
    std::cout   << "\n" << num << " shares of " << company << " sold at $" << price << std::
    endl;
    return *this;
}

Stock Stock::acquire(const char * co){
    std::strncpy(company, co, 29);
    company[29] = '\0';
    shares = 0;
    share_val = 0;
    set_tot();
    return *this; //
}

Stock Stock::acquire(const char * co, int n, double pr){
    std::strncpy(company, co, 29);
    company[29] = '\0';
    if(n < 0){
        std::cerr << "Number of shares can't be negative; "
                  << company << " shares set to 0.\n";
        shares = 0;
    }
    else{
        shares = n;
    }
    share_val = pr;
    set_tot();
    return *this; //
}

Stock Stock::update(double price){
    share_val = price;
    set_tot();
    return *this;
}

void Stock::display(){
    std::cout << std::endl << "Company: " << company << std::endl
              << "Shares: " << shares << std::endl
              << "Share value: " << share_val << std::endl
              << "Total value: " << total_val << std::endl;
}
```

```
int main(){
    using std::cout;
    using std::endl;

    //full stock
    Stock stock1("Nanosmart", 20, 12.50);
    stock1.display();
    stock1.buy(30,12.00).display();
    stock1.sell(40,13.00).display();
    stock1.update(5.00).display();

    //just company stock
    Stock stock2("Amazon");
    stock2.display();
    stock2.buy(40,100).display();
    stock2.update(150).display();
    stock2.sell(50,125).display();

    //empty stock-no company name
    Stock stock3;
    stock3.display();
    stock3.buy(5,0.55).display();
    return 0;
}
```