# CSE 333

## Lecture 16 - starting in on subclasses

**Steve Gribble**

Department of Computer Science & Engineering

University of Washington

## Administrivia

HW3 is out today

- due in two weeks
- you can work solo, or in teams of two

Your midterm is...

- on Monday May 9th
    - it covers C, C++ up to, and including, lec14
        - **DO ALL OF THE EXERCISES FROM LEC1 - LEC14!**

Section tomorrow

- details on C++ subclasses, inheritance

## Today

Go through HW3

- lots of details to understand and master

Start in on C++ inheritance

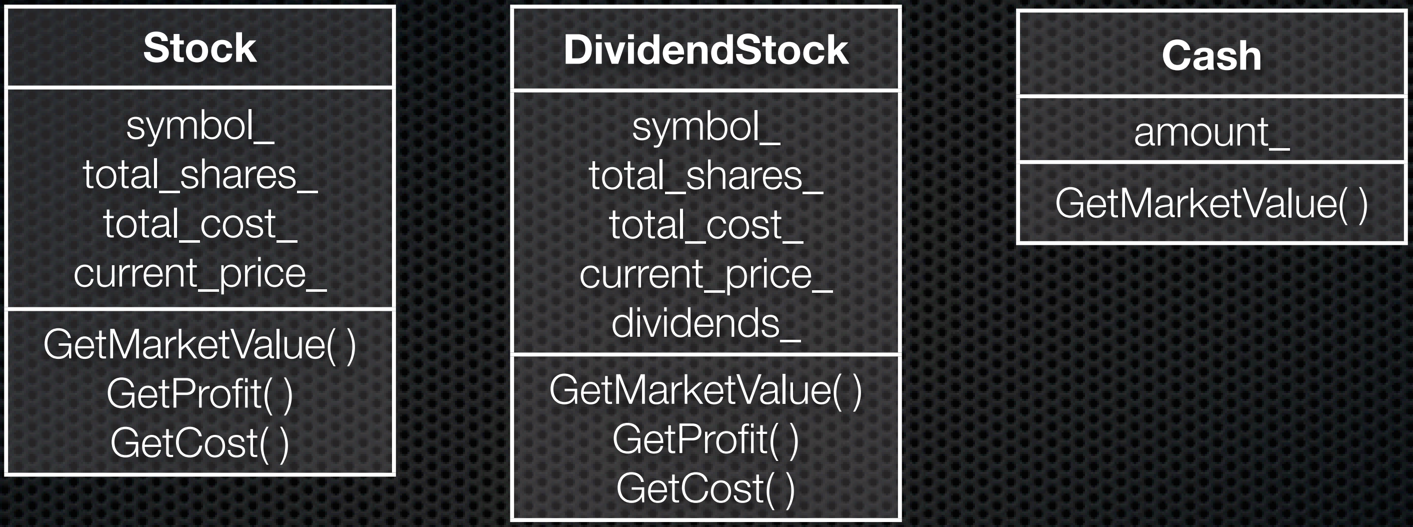
- *huge thanks to Marty Stepp for his "portfolio" case study*

## Let’s build a stock portfolio

A portfolio represents a person’s financial investments

- each asset has a cost (how much was paid for it) and a market value (how much it is worth)
    - the difference is the profit (or loss)
- different assets compute market value in different ways
    - **stock**: has a symbol (“GOOG”), a number of shares, share price paid, and current share price
    - **dividend stock**: is a stock that also has dividend payments
    - **cash:** money; never incurs profit or loss.   (hah!)

## One possible design



One class per asset type

- Problem: redundancy
- Problem: cannot treat multiple investments the same way
    - e.g., cannot put them in a single array or Vector

*see initial_design/*

## Inheritance

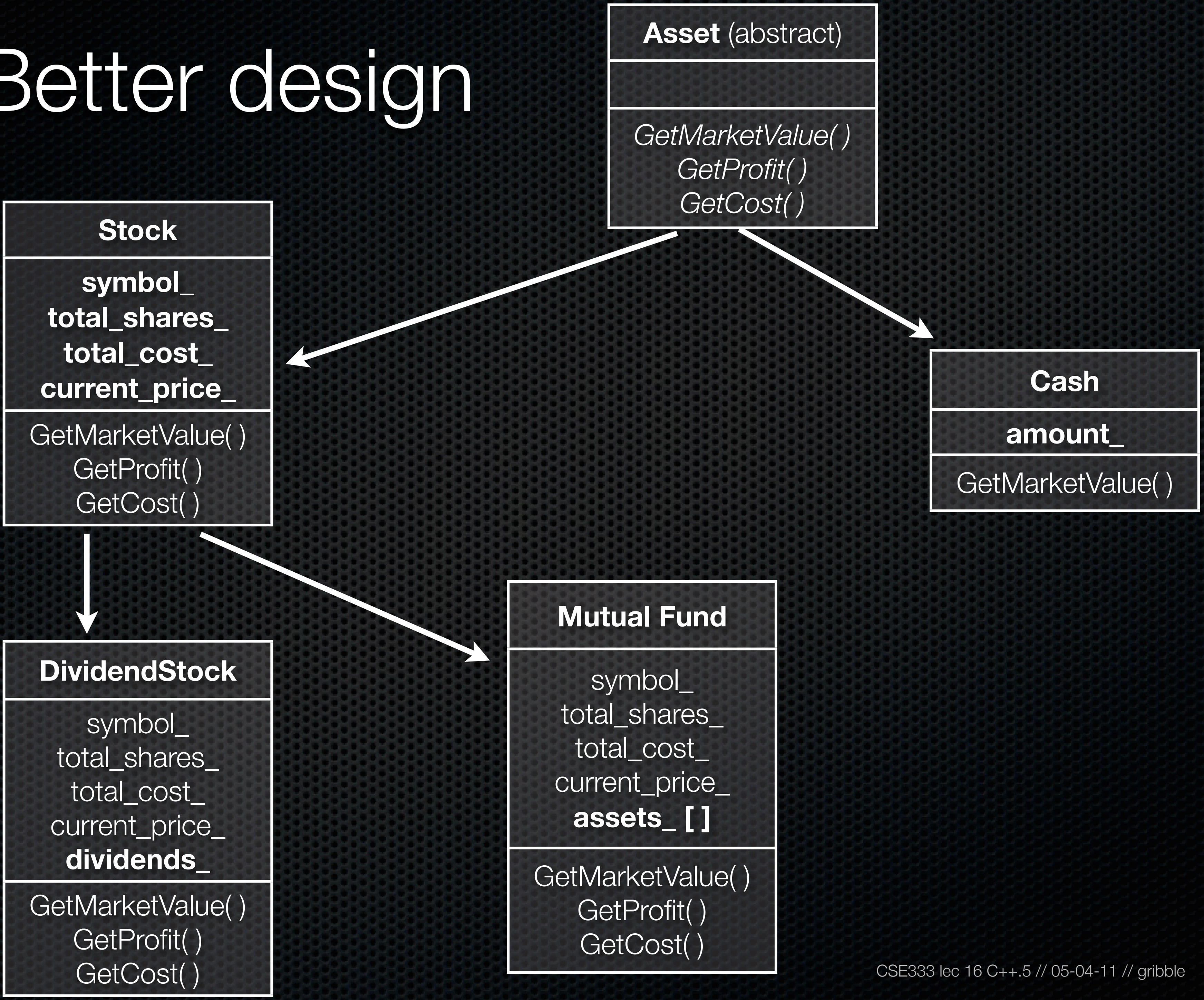
A parent-child relationship between classes

- a child (**derived** class) extends a parent (**base** class)

Benefits:

- code reuse:  subclasses inherit code from superclasses
- polymorphism
  - ability to redefine existing behavior but preserve the interface
  - children can override behavior of parent
  - others can make calls on objects without knowing which part of the inheritance tree it is in
- extensibility:  children can add behavior

# Better design



## Access specifiers

| | |
|---|---|
| **public**: | visible to all other classes |
| **protected**: | visible to current class and its subclasses |
| **private**: | visible only to the current class |

declare a member as **protected** if:

- you don't want random customers accessing them
    - you want to be subclassed and to let subclasses access them

## Public inheritance

```
#include "BaseClass.h"

class Name : public BaseClass {
  ...
};
```

- "public" inheritance
  - anything that is [*public, protected*] in the base is [*public, protected*] in the derived class
- derived class inherits **almost** all behavior from the base class
  - not constructors and destructors
  - not the assignment operator or copy constructor

*fix DividendStock in next_design/*

See you on Friday!