Inheritance

Inheritance

- By Example: Suppose we have our **Car** class and we want to have a more specialized class called "**SUV**".
  - Now, **SUV** can be considered a "**Car**" and a member of the **Car** class, but there are a number of differences.
  - Rather than develop a completely new class, called "**SUV**" from scratch, we can add to the class **Car** whatever data and methods it needs to become a **SUV**.

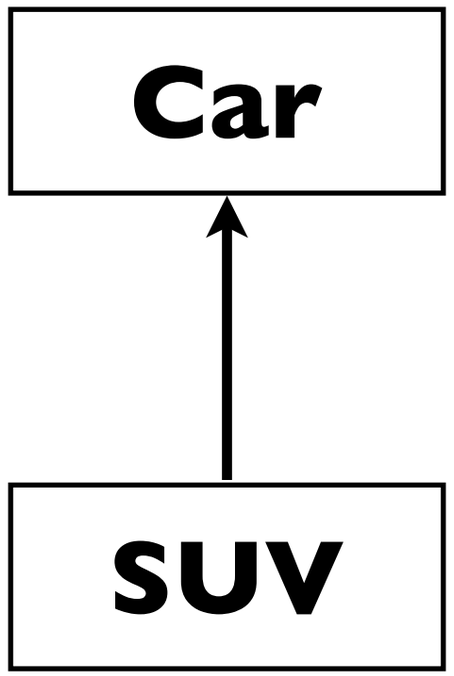
Inheritance

- Example continued: For example, the new **SUV** class might need additional data members to describe whether it has a third seat. Or how many passengers can it hold. As well as methods to operate on this data (compute price, etc.).

Inheritance

- The new resulting class is said to be inherited from the class **Car** and is called the derived class or sometimes called the subclass of the **Car** class.
- The original **Car** class is called the base class or superclass (or parent class) of **Car** class.

As Shown



Class **SUV** is derived from class **Car**

An Exercise

- Using your class **CollegeStudent** describe a derived class (Use a UML class diagram) **GraduateStudent**

Some C++ Syntax

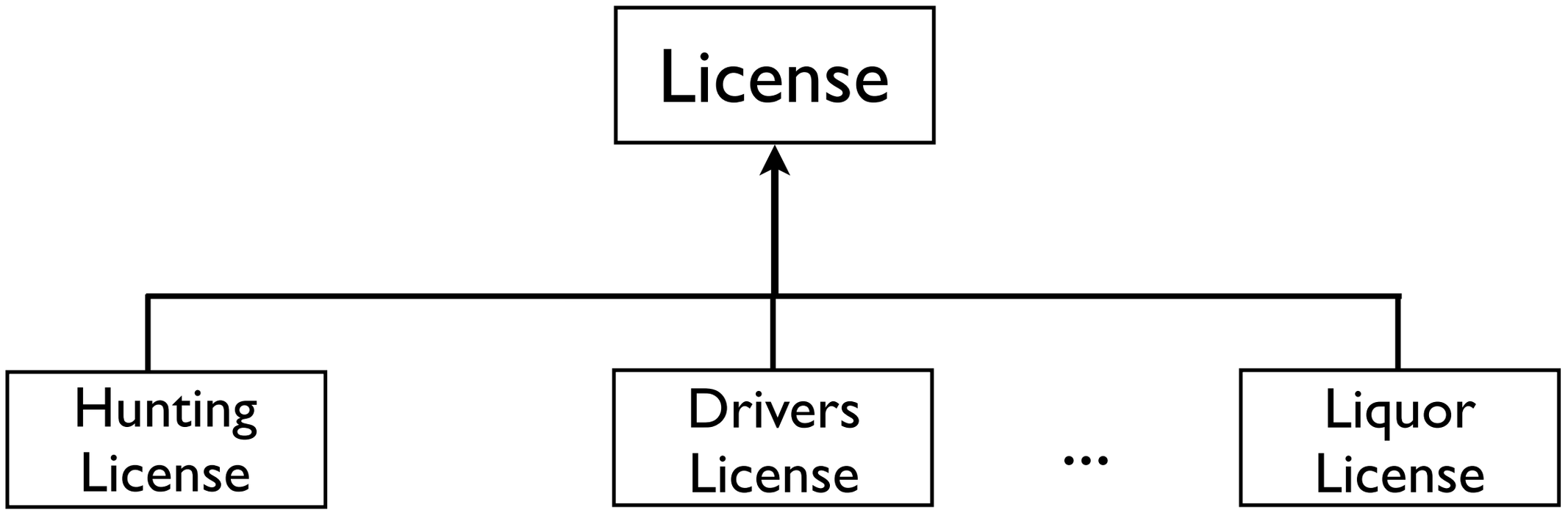
To derive **DerClass** from **BaseClass**:

```
class DerClass  :  public BaseClass {
//.......
};
```

Some Notes: This looks very similar to normal class syntax, except for : public BaseClass

The keyword **public** says that the derivation is public and BaseClass indicates that DerClass is the derived class

Can Have Multiple Inheritance



The Fundamental Property of Derived Classes

- Derived classes inherits the members of its base class and inherits the members of its ancestor classes.
  - It cannot access private members of the base class
  - The access it has to public and protected members depends on the kind of inheritance

Data Members

- Data members declared as *private* cannot be accessed outside of their class (except by friend functions), not even within a derived class
- Data members declared as *protected* can be accessed within a derived class, but are inaccessible to non-derived classes (except for friend functions).

```
class License {
public :
// ...Function Members

protected:
long int  mynumber;
string  LastName;
string FirstName;
int  Age;
int BirthYear;
int BirthMonth;
int BirthDay;
...
};
```

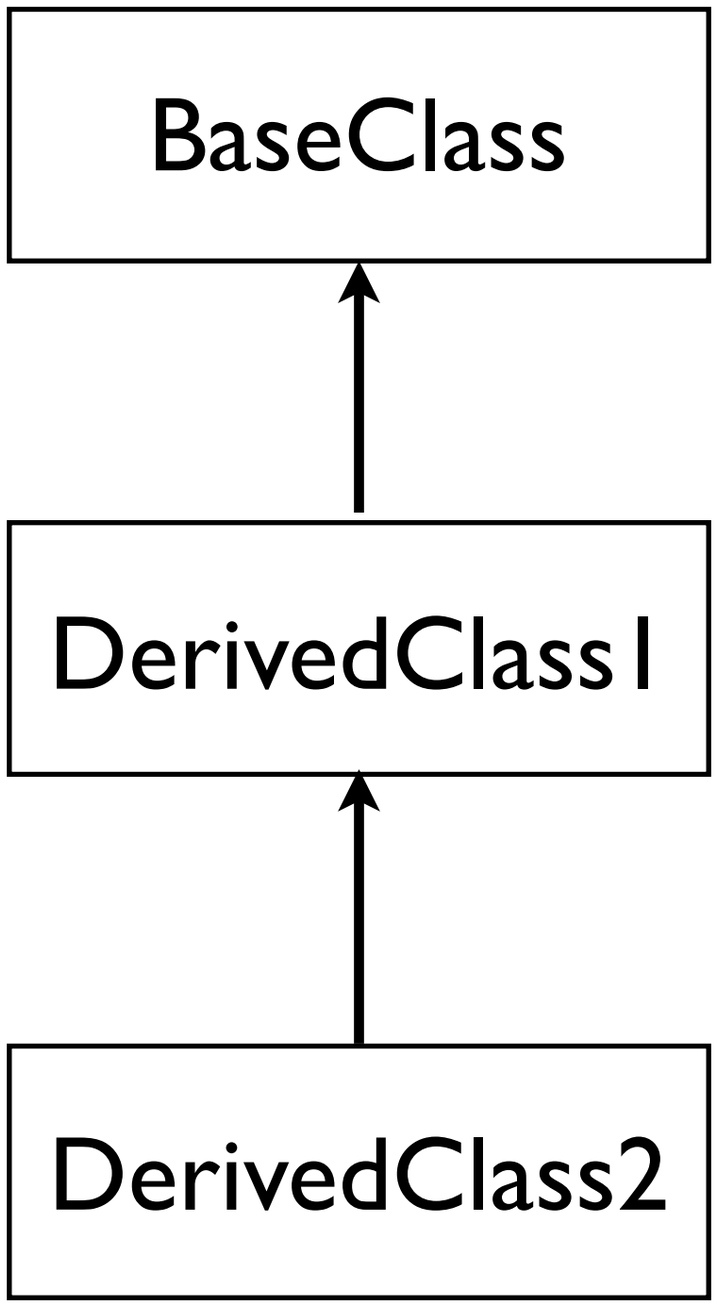
```
class DriversLicense :  public License
{
public :
// ...Function Members

protected:
int  VehicleType;   //car, truck, motorcycle, bus....
char restriction_code;
...
};
```

```
class LiquorLicense :  public License
{
public :
// ...Function Members

protected:
int   BusinessType   //bar, restaurant, store....
int ExpirationYear;
int ExpirationMonth;
...
};
```

Indirect Inheritance



Indirect Inheritance

With Multiple Inheritance