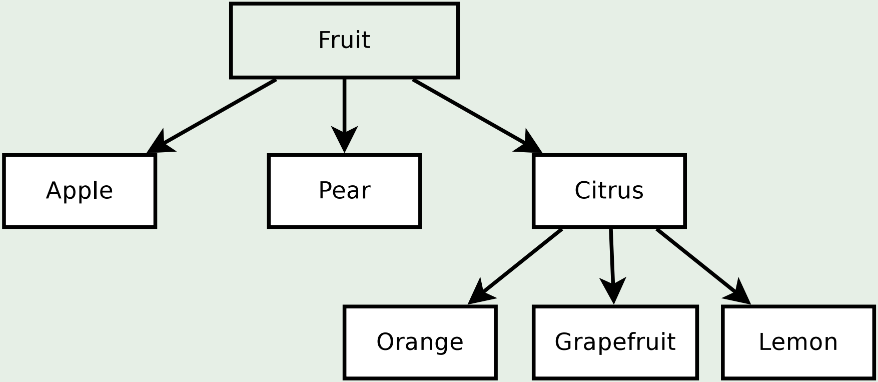
OOP: Inheritance (Derived Classes)

Example (abstract object hierarchy)



Inheritance

Definition

- child classes are derived from other parent classes
- automatically include parent's members
- inherit all the accessible members of the base class

Specifics

A derived class inherits every member of a base class except:

- its constructor and destructor
- its assignment operator
- its friends

Inheritance

Syntax

Base Class

```
class baseclass {
   protected:
   ...
   public:
      baseclass ()
      ...
};
```

Derived Class

```
class derivedclass :  public baseclass {
   ...
   public:
      derivedclass :  baseclass ()
      ...
};
```

Inheritance

Example (Matrix Base Class)

```
class matrix {
  protected:
     int rows, cols;
     double *elements;
  public:
     matrix(int r, int c);
     ~matrix();
     int get_rows();
     int get_cols();
     void fill(double value);
     matrix operator+ (const matrix &C)
};
```

Inheritance

Example (Square Matrix Derived Class)

```
class squarematrix :  public matrix {
   private:
   protected:
   public:
      squarematrix(int r, int c) :  matrix(r,c) {
         if(r!=c) std::cerr<<"not a square matrix"; exit(1);
      }
      double trace() {
         double sum(0.0);
         for(int i=0; i <rows ; i++)
         sum += elements[i*cols+i];
         return sum;
      }
};
```

Inheritance

Example

```
matrix P(5,5);
squarematrix Q(5,5);
P.fill(1.6);
Q.fill(1.6);
std::cout<<" Trace = "<<Q.trace();
```

HANDS-ON:

Come up with a derived class inheriting the matrix class as a base class.