Object Composition

- This is when one object is a member variable of another class
- This relationship is called a *has-a* relationship
Example

- Person class has-a Date object
- Let’s look at the code that implements the above
class Date
{
private:
    string month;
    int day;
    int year;

public:
    Date(string month, int day, int year)
    { setDate(month, day, year); }
    Date(){ setDate("January", 1, 1900); }
    void setDate(string month, int day, int year)
    {this->month = month; this->day = day; this->year = year;}
    string getMonth() { return month; }
    int getDay() { return day; }
    int getYear() { return year; }
};
Person Class

class Person
{
private:
    string name;
    Date dob;    // has-a Date object (Composition)
public:
    Person(string name, string month, int day, int year)
    { this->name = name; dob.setDate(month, day, year); }
    void print()
    { cout << name << "\'s birthday is on "
      << dob.getMonth() << "  " << dob.getDay() <<
      "", " << dob.getYear();
    }
};
Main Function

Person buddy("Bill Stump",
             "February", 5, 1975);

buddy.print();
Inheritance

• Classes that use inheritance are said to have an *is-a* relationship

• Examples:
  ○ Person *has-a* Date
  ○ Student *is-a* Person
  ○ Faculty *is-a* Person
Protected Data Members and Functions

• Until now, we’ve been working with two access specifications:
  - private
  - public

• Another access specification is:
  - protected
Protected

- Recall from the example last time, that Person class contained one private data member
  - `string name;`

- This meant that functions in the class Student (that is derived from Person) could not directly access Person’s private data members
  - `Student(string aName) { name = aName; }`
Protected

• Protected members of a class are just like private members, except that derived classes may access them directly
Base Access Specifications

- Recall that Student was publicly derived from Person
  
  - \texttt{class Student : public Person}

- This is called the base access specification

- We could also use private or protected
  
  - \texttt{class Student : public Person}
  
  - \texttt{class Student : protected Person}
  
  - \texttt{class Student : private Person}
Base Access Specifiers

Base class members

private: x
protected: y
public: z

private: x
protected: y
public: z

private: x
protected: y
public: z

How base class members appear in derived class

private base class

x inaccessible
private: y
private: z

protected base class

x inaccessible
protected: y
protected: z

public base class

x inaccessible
protected: y
public: z
Constructors

• When creating an object of a derived class, which constructor is called first?
  ○ The base class first
  ○ Then the derived class

• When destroying an object of a derived class, which destructor is called first
  ○ The derived class first
  ○ Then the base class