Chapter 15
Inheritance

How can we reuse our code?

Spring 2016

Key Terminology

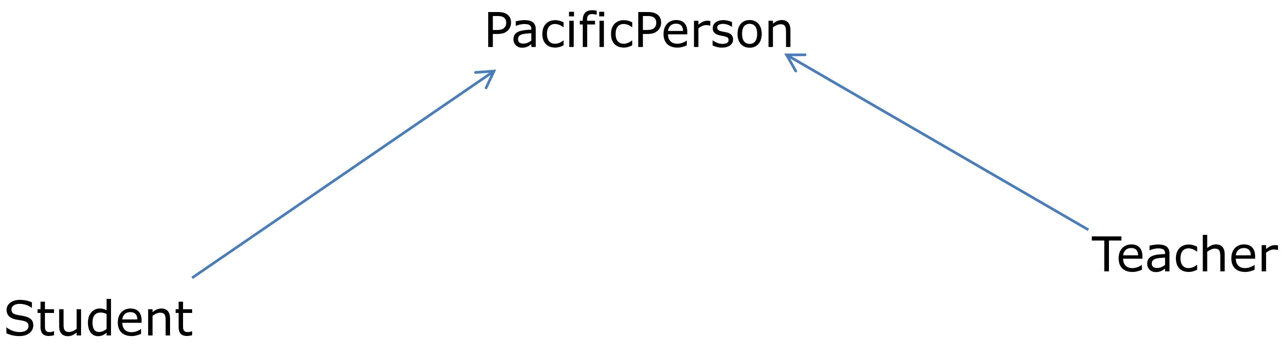
- Inheritance
- Superclass / Base class / Parent class
- Subclass / Derived class / Child class
- is-a relationship
- **Composition**
- **has-a relationship** *DNASet has-a DNA*

Inheritance

a form of reusability

- Allows a **derived** class to be based on an existing class **base** class
- The derived class inherits
  - all member variables
  - all member functions (excluding the constructors and destructor)
- The derived class ***may*** then add new functionality and new member variables

Simple Inheritance



PacificPerson

Student

Teacher

Inheritance and the is-a relationship

- A car is-a vehicle
- A rectangle is-a shape
- An athlete is-a person
- A football player is-an athlete

Simple Inheritance Example

```
class Employee
{
  public:
    Employee (string name = "");
    string getName () const;
    void setName (string);
    friend ostream& operator<<(ostream &out,
                  const Employee &cEmp);
  private:
    string mName;
};
```

HourlyEmployee

- An hourly employee is an employee that
  - earns an hourly wage rate
  - works a certain number of hours

- What additional member variables and behavior is necessary for an hourly employee?

Public Inheritance

- Every derived class object is also an object of the superclass.
- As an example, if the superclass is "Vehicle" then a subclass might be "Cars" and "Trucks." Cars inherit the members and behaviors of a Vehicle and add other behaviors and members
- Members of a subclass cannot directly access the private members of a superclass