Chapter 10
Inheritance, Polymorphism, Virtual Functions
Spring 2019
Vocabulary

- Inheritance
- Superclass
- Subclass
- Base class
- Derived class
- is-a relationship
- Composition
- has-a relationship

We will define as we go along today.
Inheritance

- Allows a new class to be based on an existing class
  - Reusability
- The new class inherits:
  - The new class then adds new functionality
  - Specialize
Simple Inheritance

Person
- mName : string
- mAge : int
+ Person(string, int)
+ getName() const : string
+ getAge() const : int

BaseballPlayer
- mHits : int
- mAttempts : int
+ BaseballPlayer(string, int, int, int)
+ getAverage() const : double

CarDealer
- mCarsSold : int
+ CarDealer(string, int, int)
+ getCommission() const : double
is-a vs has-a

- Inheritance
  - A car is-a vehicle
  - An athlete is-a person
  - A football player is-a athlete
- Composition
  - A DNASet has-a DNA
  - A Course has-a Student
OOP Vocabulary

- Inheritance is used to create an is-a relationship
class Employee
{
  public:
    Employee (string name = "", string ssn = "");
    string getName () const;
    string getSSN () const;
    void print (ostream &rcOut) const;

  private:
    string mName;
    string mSSN;
};
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? Let’s draw the UML diagram.
<table>
<thead>
<tr>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>mName : string</td>
</tr>
<tr>
<td>mSSN : string</td>
</tr>
<tr>
<td>Employee(string, string)</td>
</tr>
<tr>
<td>getName() const : void</td>
</tr>
<tr>
<td>getSSN() const : void</td>
</tr>
<tr>
<td>print(ostream &amp;) const : void</td>
</tr>
</tbody>
</table>
HourlyEmployee.h

#include "Employee.h"

class HourlyEmployee
HourlyEmployee.cpp

#include "HourlyEmployee.h"
Practice

• Grab the solution Inheritance from CS250 Public

• Rename the folder to Inheritance_PUNETID

• This contains Employee, HourlyEmployee, and a small driver.
Homework for Wednesday:

• Edit the UML diagram in Inheritance to add
  – a subclass (derived class) called SalariedEmployee for an employee that works on a yearly salary model.
  – Write the interface (.h) and implementation (cpp) for SalariedEmployee.
• Make sure this is done before next class
Practice – Coding Lab

• Look at the employees.txt file in Inheritance

• Add a member function to each class:
  
  ```cpp
  bool read(istream &rcIn);
  ```
Practice

• Create two arrays
  - ten HourlyEmployees
  - ten SalariedEmployees.

• Read the data from the data file into the appropriate array in a single loop without printing any information.

• H means hourly, S means Salaried

• Print out each employee
  - Hourly first
  - Salaried second.
Practice

• Results

Hourly

Name: Ford SSN: 234567890 Wage: 10 HoursWorked: 40 Pay: 400
Name: Goodman SSN: 345678901 Wage: 15 HoursWorked: 45 Pay: 675
Name: Smith SSN: 678901234 Wage: 22.5 HoursWorked: 40 Pay: 900

Salary

Name: Black SSN: 123456789 Salary: 32000
Name: Howell SSN: 456789012 Salary: 44000
Name: Powell SSN: 567890123 Salary: 50000