ECE122

March 29, 2005
Another Example of Inheritance

• I want to write classes of medical doctors. One is family practitioner, one surgeon.
• How many classes do I need?
• What are they?
• Can I use inheritance?
• What should I put in super class?
• What should I put in the sub class?
Is there a IS-A relationship?

• Both Family practitioner and surgeon are specific kind of medical doctors.
• So I can use Inheritance to simplify my design and coding!
How many classes do I need? What are they?

• I need three classes.
• Class Doctor is a super class that contains the common instance variables and methods that are shared among all doctors.
• Class FamilyPractitioner is a subclass of class Doctor.
• Class Surgeon is a subclass of class Doctor.
What should I put in the super class?

• Common instance variables and methods shared by all kinds of medical doctors.
• Instance variable: boolean worksAtHospital
• Method: void treatPatient()
• Now show me the class
Super class: Doctor

public class Doctor
{
    boolean workAtHospital;
    public void treatPatient()
    {
        //default implementation: perform a checkup
    }
}
What is special about FamilyPractitioner class?

• It needs to distinguish whether this doctor makes house calls. So we add one instance variable: boolean makesHouseCalls

• A family practitioner gives healthy lifestyle, and nutrition advice. So we add one method: giveNutritionAdvice().

• Do we need to override treatPatient() method? No. The super class’s method implements just what a family practitioner does.
public class FamilyPractitioner extends Doctor
{
    boolean makeHouseCalls;
    public void giveNutritionAdvice()
    {
        //Give nutrition Advice to all family members.
    }
}
What is special about Surgeon?

• A surgeon treat patient differently. He/she doesn’t just do a check up, as defined as a default behavior for all doctors in the super class. So a Surgeon class need to override this method: treatPatient()

• A surgeon need a surgical procedure to get his/her job done. So we need to add another method, surgicalProcedure().
public class Surgeon extends Doctor
{
    public void treatPatient()
    {
        //Perform Surgery
    }
    public void surgicalProcedure()
    {
        //make incision ...
    }
}
Demo

• Run FamilyPractitioner and Surgeon.
• Figure out whose methods get called.
Composition

- Composition reflects a HAS-A relationship among classes/objects.
- A bathroom has a tub.
- Neither bathroom is a tub, nor tub is a bathroom.
- But bathroom contains a tub. This is a HAS-A relationship.
- A bathroom is composed of many parts, like tub, sink, etc.
- A Bathroom class/object contains other objects, like Tub, Sink, etc.
Example: Bathroom HAS A Tub

public class Tub
{
   //Define Tub
}

public class Bathroom
{
   Tub bathtub; //Bathroom HAS A Tub
   //Other definition
}
Tub class definition

```java
public class Tub {
    private String style;
    public String getStyle() {
        return style;
    }
    public void setStyle(String style) {
        this.style = style;
    }
}
```
Tub object is created inside Bathroom class constructor

Bathroom(String style)
{
    bathtub = new Tub();
    bathtub.setStyle(style);
}
Bathroom class pass method calls to Tub class

```java
public String getTubStyle()
{
    return bathtub.getStyle();
}

public void setTubStyle(String style)
{
    bathtub.setStyle(style);
}
```
Demo with Eclipse