CSE 143 – Section AA

June 26, 2003

Test procedures

- Place a static method in each class, just for testing it.
  - No special name; could even be main().
  - Even simple tests are helpful
  - Run the test method every time the class is modified

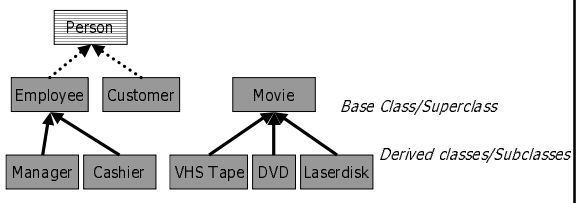
```
class BankAccount {
 /** A method to test out some of the BankAccount operations */
  public static void test() {
    BankAccount myAccount = new BankAccount("Joe Bob");
    myAccount.deposit(100.00);
    myAccount.deposit(250.00);
    myAccount.withdraw(50.00);
    System.out.println(myAccount);  // uses myAccount.toString()
  }

} // end of BankAccount
```

Relationships

- Specialization
  - Relationship between more general and more specific ideas
    - Ford Escort IS-A Car
    - Ford Taurus IS-A Car
    - Car IS-A Vehicle
- Composition
  - Relationship between different things
    - Car HAS-A engine
    - Engine HAS-A pistons
    - Note: Student HAS-A Teacher

IS-A/HAS-A in Java

- HAS-A is encoded as the contents of a class
  - Most of what you've done is "HAS-A"
    - Student HAS-A age
    - Student HAS-A name
- IS-A is encoded as relationships between classes
  - Lets us group classes
  - Lets us partially reuse implementations



IS-A == Inheritance

- When defining a class, we can say it "extends" an existing class
  - You've actually been "extending" a class all the time. If no class is specified, a class extends Object
- A subclass can be used anywhere a superclass is specified
- A subclass uses the method implementations and instance variables of the superclass
  - Can redefine parent methods
  - Some restrictions on access to instance variables
- Note: interfaces are a special case of class inheritance

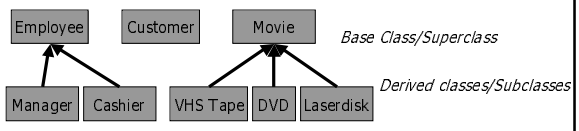
Example

```
class Movie {
  …
   public String getInfo() {
       return title+length;
  }
}
class VHSTape extends Movie {
  …
}
class DVD extends Movie {
  …
  public String getInfo() {
       return title+length+extraFeatures;
  }
}
```

Inheritance and polymorphism

```
Movie allMovies[] = new Movie[2]
allMovies[0]      = new DVD(…)
allMovies[1]      = new VHSTape(…)
….
for (i=0;i<allMovies.length;i++) {
  System.out.println(allMovies[I].getInfo());
}
```

Video store database



- Design a video store rental database
  - Find some other classes and IS-A relationships between them
  - Specify important HAS-A relationships
  - Specify important objects