Announcements

- P5 due today at 3pm
- Prelim 3 on 4/20 (T) 7:30-9pm
- Correction to notes 4/13, method roll() in TrickDice:
  ```java
  int r = (int) (Math.random() * …) …;
  if (r <= getSides())
    setTop(r);
  else
    setTop(weightedSide);
  ```

Inheritance

- Allows programmer to **derive** a class from an existing one
- Existing class is called the **parent class**, or **superclass**
- Derived class is called the **child class** or **subclass**
- The child class **inherits** the (public) members defined for the parent class
- Inherited trait can be **accessed** as though it was **locally declared (defined)**

Reserved word **super**

Invoke constructor of superclass

```
super(parameter-list);
```

parameter-list must match that in superclass’ constructor

Overriding methods

- Subclass can **override** definition of inherited method
- New method in subclass must have same signature as superclass (but has different method body)
- **Which method gets used??**
  *The object that is used to invoke a method determines which version is used*
- Method declared to be **final** cannot be overridden
- Do not confuse **overriding** with **overloading**!

Different hierarchy for different case

```
Vehicle
  ↓
CargoType
  ↓
Car
  ↓
CargoPlane
  ↓
Vehicle
  ↓
Car
  ↓
Plane
  ↓
PassengerType
```
Single inheritance

```
Vehicle
  ┌── CargoType
  │   PassengerType
  │       Truck
  │       Car
  │       Plane
```

NO!

Single inheritance—one parent only

Important ideas in inheritance

- Use different hierarchies for different problems
- Single inheritance
- Keep common features as high in the hierarchy as reasonably possible
- Use the superclass’ features as much as possible
- Inherited features are continually passed down the line
- “Inherited” ⇒ ”can be accessed as though declared locally”
  (private variables in superclass exists in subclasses; they just cannot be accessed directly)

Accessing members in superclass

```
super

From constructor in subclass, call superclass’ constructor
Access superclass’ version of a overridden method. E.g.:

    super.toString()
```

Polymorphism

“Have many forms”

- A polymorphic reference refers to different objects (related through inheritance) at different times